

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

# **Aplikace pro pořizování 3D fotografií na platformě Android**

## **3D Photography Application for Android Devices**

## Zadání diplomové práce

Student: **Bc. Lukáš Janoška**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Aplikace pro pořizování 3D fotografií na platformě Android**  
**3D Photography Application for Android Devices**

Jazyk vypracování: čeština

### Zásady pro vypracování:

Cílem práce je tvorba knihovny a aplikace pro tvorbu 3D fotografií a jejich zobrazování různými vizualizačními technikami (anaglyf, side-by-side, apod.) na platformě Android. Hlavní část práce bude zaměřena na metody, použité pro (polo)automatické spojení dvou (popř. více) snímků do výsledného obrazu včetně korekce otočení a náklonu a indikace vhodného okamžiku pro pořízení dalšího snímku.

1. Prostudujte současné techniky pro tvorbu anaglyfů a 3D fotografií.
2. Zjistěte, jaké aplikace již v dané oblasti existují a jaké možnosti poskytují.
3. Implementujte knihovnu s vhodnou metodou automatického spojení snímků, alespoň 2 výstupní formáty pro reprezentaci stereoskopického snímku a aplikaci, která bude provádět snímání v ručním či (polo)automatickém režimu na platformě Android.
4. Výslednou aplikaci otestujte a srovnajte Vaše výsledky s dostupnými alternativami (kvalita, rychlost zpracování, atd.).

### Seznam doporučené odborné literatury:

- [1] E. Dubois: A Projection Method to Generate Anaglyph Stereo Image, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001, DOI: 10.1109/ICASSP.2001.941256.
- [2] A. A. Krupev, A. A. Popova: Ghosting Reduction and Estimation in Anaglyph Stereoscopic Images, Proceedings of IEEE International Symposium on Signal Processing and Information Technology, 2008, DOI: 10.1109/ISSPIT.2008.4775719.
- [3] I. Ideses, L. Yaroslavsky: New Methods to Produce High Quality Color Anaglyphs for 3-D Visualization, Lecture Notes in Computer Science Volume 3212, 2004, pp 273-280.
- [4] Meier, R. Professional Android 2 Application Development. Wrox Press Ltd., 2010. ISBN: 978-0-47056-552-0.
- [5] John C. Russ: The Image Processing Handbook, Sixth Edition, CRC Press, 2011, ISBN 1439840636.

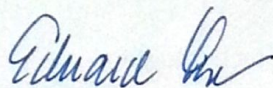


Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Pavel Moravec, Ph.D.**

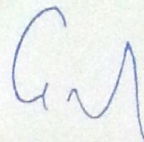
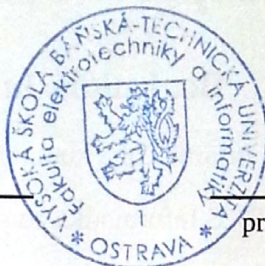
Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



---

doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



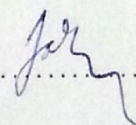
---

prof. RNDr. Václav Snášel, CSc.  
děkan fakulty



Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární  
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 1. dubna 2016

..........



Rád by som na tomto mieste poďakoval všetkým, ktorý mi boli ochotní s diplomovou prácou pomôcť, predovšetkým vedúcemu práce Ing. Pavlovi Moravcovi Ph.D.

## **Abstrakt**

Cieľom tejto diplomovej práce bolo vytvorenie knižnice a aplikácie na platforme Android pre tvorbu 3D fotografií a ich zobrazovanie rôznymi vizualizačnými technikami. V práci je rozobraná problematika stereoskopie a snímania 3D fotografií a zároveň podrobne popísaný návrh a implementácia aplikácie Camera3D a knižnice v nej použitej. Hlavná časť tejto práce je zameraná na popis vytvorenia metódy polo-automatického a manuálneho spojenia 3D snímky vrátane korekcie otočenia a naklonenia, a na indikáciu vhodného okamžiku pri snímaní druhej fotografie stereoskopického páru.

**Kľúčová slova:** diplomová práca, Android, fotografia, stereoskopia, 3D, Anaglyf, Side-By-Side

## **Abstract**

The goal of this master thesis is to make a library and application for creation of 3D photographs on Android platform and their displaying by various visualisation techniques. The thesis analyses the issue of stereoscopy and taking 3D photographs and also describes in detail the draft and implementation of Camera3D application and the library used in it. The main part of the thesis is focused on describing the creation of a method of semi-automatic and manual joint of a 3D picture including correction of turning and inclination and on indication of the right moment during taking of the second photo from the stereoscopic pair.

**Key Words:** master thesis, Android, photography, stereoscopy, 3D, Anaglyph, Side-By-Side



# Obsah

<b>Zoznam použitých skratiek a symbolov</b>	<b>9</b>
<b>Zoznam obrázkov</b>	<b>10</b>
<b>Zoznam tabuliek</b>	<b>11</b>
<b>1 Úvod</b>	<b>13</b>
<b>2 Použité technológie</b>	<b>14</b>
2.1 Dalvik a ART . . . . .	14
2.2 Java Native Interface . . . . .	14
2.3 Snímanie fotografií na platforme Android . . . . .	14
<b>3 Stereoskopia</b>	<b>16</b>
3.1 Prirodzené binokulárne videnie . . . . .	16
3.2 Podnety pre vnímanie hĺbky priestoru . . . . .	17
3.3 Umelé stereoskopické videnie . . . . .	18
3.4 Stereofotografia . . . . .	18
3.5 Určenie vhodnej miery priestorovosti snímky . . . . .	20
3.6 Spôsoby zobrazenia 3D snímky . . . . .	21
3.7 Snímanie stereoskopického páru . . . . .	25
3.8 Formáty uloženia 3D snímky . . . . .	26
<b>4 Popis metód automatického spojenia stereoskopického páru</b>	<b>28</b>
<b>5 Porovnanie existujúcich aplikácií z kategórie stereoskopického snímania</b>	<b>29</b>
5.1 Požadované vlastnosti a chovanie aplikácií . . . . .	29
5.2 Popis a testovanie aplikácií . . . . .	29
<b>6 Návrh aplikácie Camera3D</b>	<b>32</b>
6.1 Iteratívny návrh používateľského rozhrania . . . . .	32
6.2 Funkčné požiadavky . . . . .	32
6.3 Nefunkčné požiadavky . . . . .	41
<b>7 Analýza</b>	<b>42</b>
<b>8 Implementácia výpočtovej knižnice</b>	<b>43</b>
8.1 Metódy knižnice napísané v jazyku C . . . . .	43
8.2 Metódy napísané v jazyku Java okrajovo používajúce knižnicu . . . . .	44

<b>9 Implementácia aplikácie Camera3D</b>	<b>47</b>
9.1 Hlavná aktivita . . . . .	47
9.2 Použitie fragmentov . . . . .	47
9.3 Snímanie fotografií . . . . .	48
9.4 Indikácia vhodného okamžiku snímania - zobrazenie senzorickej navigácie . . . .	51
9.5 Ovládanie pohybu guľičiek na základe senzorov . . . . .	51
9.6 Riešenie problémov spôsobených nízkou kapacitou operačnej pamäte mobilných zariadení . . . . .	52
9.7 Uloženie 3D snímky na kartu SD . . . . .	53
9.8 Prezeranie vytvorených snímok v galérii . . . . .	55
9.9 Varovné dialógy . . . . .	56
<b>10 Testovanie aplikácie Camera3D</b>	<b>58</b>
<b>11 Porovnanie aplikácie Camera3D s existujúcimi aplikáciami</b>	<b>60</b>
11.1 Indikácia vhodného okamžiku pri snímaní druhej fotografie . . . . .	60
11.2 Najvyššie rozlíšenie snímok . . . . .	61
11.3 Polo-automatické spojenie snímok . . . . .	61
<b>12 Záver</b>	<b>63</b>
<b>Literatúra</b>	<b>64</b>
<b>Prílohy</b>	<b>65</b>
<b>A Zdrojové kódy</b>	<b>66</b>
<b>B Ukážkové obrázky vyrobené a upravené aplikáciou Camera3D</b>	<b>67</b>



## Zoznam použitých skratiek a symbolov

JNI	– Java native interface
JVM	– Java virtual machine
ART	– Android Runtime
AOT	– Ahead-Of-Time
JIT	– Just-In-Time
NDK	– Native development kit
SURF	– Speeded-Up Robust Features
SIFT	– Scale invariant feature transform

## Zoznam obrázkov

1	Vľavo správna pozícia fotoaprátu, vpravo nesprávna. Prevzaté a upravené z [23] .	19
2	Stereoskop, Google Cardboard. Prevzaté z [16] [27] . . . . .	23
3	Anaglyf test . . . . .	24
4	Stereo sánky prevzaté z [13] . . . . .	26
5	Návrh UI . . . . .	32
6	Rotácie zariadenia na osách X,Y,Z . . . . .	33
7	Zobrazenie pohybu zariadenia na ose X . . . . .	34
8	Zobrazenie pohybu zariadenia na osách X, Y, Z . . . . .	35
9	Nevhodné naklonenie na ose X . . . . .	36
10	Nevhodné naklonenie na ose Y . . . . .	36
11	Nevhodné naklonenie na ose Z . . . . .	36
12	Vľavo anaglyf, vpravo Side-By-Side . . . . .	37
13	Zobrazenie prvej polo priehľadnej snímky . . . . .	38
14	Manuálna korekcia pozície snímok . . . . .	39
15	Nevhodne vyrotovaný stereoskopický pár, vytvorený snímaním na výšku zariadenia	39
16	Vytvorený anaglyf s korekciou rotácie . . . . .	40
17	Rozloženie layoutu pri snímaní . . . . .	50
18	Dialóg progresu procesu . . . . .	55
19	Varovný dialóg . . . . .	56
20	SAMSUNG S4 mini - snímame prvej fotografie . . . . .	58
21	SONY Xperia M - snímame druhej fotografie . . . . .	59
22	SAMSUNG S4 mini - funkcia polo-automatického spojenia . . . . .	59
23	SAMSUNG S3 - vytvorenie anaglyfu . . . . .	59
24	3D Steroid grafické zobrazenie pozície na ose X . . . . .	61
25	Camera3D indikácia vhodného okamžiku snímania pomocou údajov zo senzorov na osách X,Y,Z . . . . .	61
26	Funkcia polo-automatického spojenia snímok . . . . .	62
27	Vľavo:bez použitia metódy polo-automatického spojenia, vpravo: s použitím me- tódy . . . . .	62
28	Ukážka anaglyfu . . . . .	67
29	Ukážka Side-By-Side 1. . . . .	68
30	Ukážka Side-By-Side 2. - vytvorená na výšku . . . . .	68



## Zoznam tabuliek

1	Porovnanie existujúcich aplikácií . . . . .	30
2	Výpis funkcií aplikácie Camera3D . . . . .	60

## Zoznam výpisov zdrojového kódu

1	Vyvolanie Intentu na zachytenie snímky [21] . . . . .	15
2	Deklarovanie natívnej metódy calculateAnaglyph() . . . . .	43
3	Kód natívnej metódy calculateAnaglyph() . . . . .	43
4	Uvoľnenie pixelov z pamäte . . . . .	44
5	Vzájomný posun dvoch CustomView . . . . .	45
6	Zistenie šírky a výšky bitmáp . . . . .	45
7	Práca s rozmermi bitmáp . . . . .	46
8	Práca v jave s fragmentmi . . . . .	47
9	Overenie existencie fotoaparátov . . . . .	48
10	Ukážka časti Android Manifestu . . . . .	48
11	Vytvorenie inštancie triedy Camera . . . . .	49
12	Vytvorenie náhľadovej triedy . . . . .	49
13	Snímanie fotografie . . . . .	50
14	Pridanie sensorsView1 do sensorsContaineru . . . . .	51
15	Implementácia metódy onSensor() . . . . .	51
16	Pohyb guľičky . . . . .	51
17	Senzory pri snímaní druhej fotografie . . . . .	52
18	Pridelenie väčšieho objemu operačnej pamäte v manifeste . . . . .	52
19	Štandardné zníženie kvality snímok . . . . .	53
20	Vlastný algoritmus pre zníženie rozlíšenia snímok . . . . .	53
21	Vytvorenie zložky . . . . .	54
22	Vytvorenie časového razítka . . . . .	54
23	Vytvorenie a zápis mediálneho súboru . . . . .	54
24	Metódy počas a po dokončení vlákna . . . . .	55
25	Otvorenie galérie . . . . .	56
26	Definícia dialógu . . . . .	56
27	Nastavenie čo sa stane po stlačení tlačidla OK . . . . .	57
28	Nastavenie čo sa stane po stlačení tlačidla Cancel . . . . .	57
29	Orezanie bitmáp v prospech spojenia snímok . . . . .	66

# 1 Úvod

Android je v súčasnosti jedným z popredných mobilných operačných systémov. Softwarová základňa pre Android je rozsiahla a obsahuje nepreberné množstvo aplikácií. Jednou zo základných kategórií sú aplikácie umožňujúce snímanie fotografií, ktoré by sa dali rozdeliť do viacerých skupín podľa účelu. Aplikácií pre snímanie 3D fotografií v súčasnej dobe existuje niekoľko a sú zamerané na jednoduché vytváranie 3D snímok, bez hlbšej snahy pomôcť používateľovi s vytvorením perfektnej priestorovej snímky. V tejto práci sú existujúce aplikácie analyzované a vzájomne porovnané. Táto práca sa zaoberá vývojom Android aplikácie a knižnice pre snímanie 3D fotografií, ktorej bol udelený názov Camera3D, pričom pri základnom návrhu aplikácie bolo prihliadané na funkcionality existujúcich aplikácií, ktorá bola podrobne analyzovaná. Hlavný dôraz pri návrhu aplikácie Camera3D bol kladený na vylepšenie vybraných funkcií z prieskumu a predstavenie nových funkcií, vďaka čomu by aplikácia získala náskok pred konkurenciou vo svojej kategórii. Už pri prvom letmom prieskume existujúcich aplikácií bolo jasné, že implementovanie funkcie indikácie vhodného okamžiku pomocou rotačných senzorov pri snímaní druhej snímky stereoskopického páru, by bolo veľkým prínosom do kategórie aplikácií stereoskopického snímania fotografií. Ďalšou kľúčovou a prínosnou metódou je implementovanie polo-automatického spojenia 3D snímky, ktoré bude tiež podrobne rozobrané v tejto práci. Na začiatku práce sú popísané niektoré z použitých technológií pri vývoji aplikácie, nasledované popisom problematiky stereoskopie a snímania 3D fotografií. V práci sú pre ukážku v krátkosti popísané aj metódy automatického spojenia stereoskopického páru s automatickou detekciou zhodných bodov. Na záver je aplikácia testovaná na reálnych zariadeniach, je porovnaná s v súčasnosti dostupnými aplikáciami a sú vyrobené 3D snímky dostupné k nahliadnutiu v prílohe diplomovej práce.



## 2 Použité technológie

Následuje popis niektorých vybraných technológií a štandardných programátorských postupov, ktoré boli použité neskôr v tejto práci.

### 2.1 Dalvik a ART

Dalvik je v informatike názov virtuálneho stroja od spoločnosti Google, ktorý v systéme Android vytvára prostredie pre beh aplikácie napísanej v jazyku Java. Jedná sa o JIT kompilátor. Programy pre Android sú prevážne písané v programovacím jazyku Java. Sú kompilované do bajtkódu pre Java Virtual Machine. Ten je následne preložený do Dalvik bajtkódu a uložený v .dex (Dalvik Executable) a .odex (Optimized Dalvik Executable) súboroch. Pojmy odex a de-odex sa používajú v spojení s konverziou bajtkódu. Kompaktný formát Dalvik Executable je navrhnutý pre systémy, ktoré sú obmedzené pamäťovou alebo výkonovou kapacitou. V systéme Android 4.4 "KitKat" bol v roku 2013 uvedený ako úkážka nástupca Dalviku a to Android Runtime (ART) a vo verzii Android 5.0 "Lollipop" v roku 2014 už ART úplne nahradil pôvodný virtuálny stroj Dalvik. [12] ART využíva spracovanie AOT, takže sa bajtkód predkompiluje do zariadenia vo chvíli inštalácie. Výhodou je rýchlosť spúšťania aplikácií, ktorá je takmer rovnaká ako pri kľúčových predinštalovaných aplikáciách. Na druhej strane je treba počítať s dlhším časom inštalácie a väčšou pamäťovou náročnosťou inštalovaných aplikácií zhruba o 10 až 20 percent. [19]

### 2.2 Java Native Interface

JNI – je rozhranie umožňujúce prepojiť kód bežiaci na virtuálnom stroji Javy s natívnymi programami a knižnicami napísanými v iných jazykoch - napr. C, C++, Assembler, atp., ktoré sú skompilované pre určitý hardware, prípadne operačný systém. Jedná sa o akýsi mostík, pomocou ktorého je možné dostať sa za hranice virtuálneho stroja. Volanie natívnych metód vyžaduje podporu kompilátoru. Android používa svoj vlastný prekladač Dalvik. JNI nie je obmedzené len na jazyk Java. Ako také je toto rozhranie súčasťou virtuálneho stroja, je teda možné k nemu pristupovať z ľubovôlej jazyka prekladaného do javovského bytekódu a spúšťaného na platforme JVM, Dalvik alebo ART. [20]

### 2.3 Snímanie fotografií na platforme Android

Na jednoduché snímanie fotografií je možné použiť vstavanú základnú aplikáciu Androidu. Spôsob delegovania činností do iných Android aplikácií je pomocou použitia Intentu (úmyslu), ktorý popisuje čoho je potreba dosiahnuť. Tento proces pozostáva z troch častí. Z Intentu ako takého, z výzvy na spustenie externej Activity, a z kódu, ktorý sa postará o vrátené obrazové dáta vo chvíli, keď sa focus znova vráti do našej Activity.

Následuje zdrojový kód popisujúci funkciu, ktorá vyvoláva Intent na zachytenie snímky1.

---

```
static final int REQUEST_IMAGE_CAPTURE = 1;

private void dispatchTakePictureIntent() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
    }
}
```

---

Výpis 1: Vyvolanie Intentu na zachytenie snímky [21]

## 3 Stereoskopia

### 3.1 Prirodzené binokulárne videnie

Zdravý človek, ktorý má správne vyvinuté oči a jeho nervová sústava mu z nich umožňuje správne spracovávať signály, vníma svet binokulárne. Binokulárne videnie znamená videnie oboma očami simultánne, tak, že sa obrazy spoja v jeden a v prospech vnímania hĺbky priestoru. Nie je vrodené, ale u každého jedinca sa vyvíja od narodenia až do jedného roku života. K jeho upevňovaniu dochádza do šiesteho až ôsmeho roku. Rozvoj prebieha paralelne v troch zložkách: optickej, motorickej a senzorickej. Ide o prirodzenú ľudskú vlastnosť. Pri pohľade každým okom samostatne, je človek schopný vnímať obraz jednotlivo z každého oka pod mierne odlišným uhlom vďaka horizontálnemu posunu očí. Pri pohľade oboma očami sú tieto obrazy spojené v jeden (tzv. fúzia obrazu) a je umožnené vnímanie hĺbky obrazu. Zorné polia oboch očí zaberajú odlišnú oblasť. Tieto oblasti sa však prekrývajú. Stereoskopické videnie je umožnené práve v oblasti tohto prieniku.

Ľudské oči sú od seba horizontálne vzdialené približne 6,5 cm. Binokulárna disparita je pre väčšinu ľudí dominantným podnetom pri vnímaní hĺbky a priestorovosti obrazu, avšak ak sú ostatné podnety prezentované nesprávne, môžu mať negatívny dopad na celkový priestorový vnem. Pri ostrení na určitý bod v priestore sa oči natočia tak, aby sa ich zorné osi v tomto bode pretínali. Pokiaľ k tomuto pretnutiu nedochádza, je tento patofyziologický jav označovaný ako strabismus (ľudovo škúlenie). Obraz sledovaného bodu sa premietne do identického miesta oboch sietníc. Mozog následne prevedie fúziu mierne odlišných obrazov a je tak vyvolaný jediný optický vnem. Človek vníma hĺbku na základe porovnávania s ďalšími objektmi, a preto je problém určiť vzdialenosť osamoteného objektu v prázdnej scéne. Mozog jednoducho nemá s čím porovnávať.[1, 2]

#### 3.1.1 Podmienky pre vznik jednoduchého binokulárneho videnia

Podmienok pre dokonalé binokulárne videnie je niekoľko. V prvom rade musí byť nenarušená správna funkcia oboch očí. Nesmie existovať žiadna vada, ktorá by videnie jedného, alebo oboch očí, obmedzovala. Je nutné, aby bol nervový systém v oblasti zrakovej dráhy v plnej funkčnosti, a tak isto aj kôrové centrá v mozgu. Ďalšími predpokladmi pre stereopsiu sú centrálna fixácia, normálna zraková ostrosť, pohyblivosť očí vo všetkých smeroch s neporušenou motorickou zložkou, normálna rektálna korešpondencia a mozgová interpretácia videného obrazu. [4]

#### 3.1.2 Stupne binokulárneho videnia

V tejto kapitole bolo čerpané zo zdrojov[3]. Binokulárne videnie je zložitý komplex procesov v zrakovom systéme a v mozgu. Tento komplex je možné rozdeliť na nasledujúce zložky:

- Simultánne videnie - vnímanie dvoch neidentických objektov každým okom zvlášť v jednom subjektívnom pohľadovom smere.
- Fúzia - schopnosť spojenia takmer totožných obrazov v jeden vnem.
- Stereoskopické videnie - schopnosť vytvoriť hĺbkový vnem, pri ktorom vzniká trojrozmerný obraz.

Ďalej je fúzia delená na centrálnu a motorickú. Centrálna fúzia zaistuje spojenie dvoch monokulárnych vnemov bez núteného pohybu očí. Pri motorickej fúzii sú využívané fúzne rezervy zrakového aparátu. Centrálna fúzia je vždy previazaná fúziou motorickou, avšak motorická fúzia nemusí obsahovať centrálnu fúziu.

### 3.2 Podnety pre vnímanie hĺbky priestoru

Informácie v tejto podkapitole boli čerpané z [5]. Existuje celý rad podnetov, ktoré ľudský vizuálny systém používa k vnímaniu hĺbky a priestorovosti. Nasleduje výpis niektorých podstatných kľúčov. Existujú aj ďalšie kľúče podstatné pri vnímaní hĺbky priestoru, ktoré však v tejto práci nie sú uvádzané, pretože nie sú prítomné v 2D snímkach:

#### 3.2.1 Monokulárne kľúče

Jedná sa o podnety, ktoré je možné rozoznať pozorovaním pomocou jedného zdravého ľudského oka.

- Perspektíva - Čím sú objekty ďalej, tým sa viac zmenšujú a rovnobežné línie konvergujú v diaľke.
- Veľkosti známych predmetov - Očakávanie, že niektoré objekty sú menšie než iné. Ak sa slon a šálka čaju objavia v rovnakej veľkosti, potom existuje predpoklad, že slon je ďalej.
- Detail - Blízke objekty sú viditeľné detailne, vzdialené menej.
- Interpozícia - Predpokladá sa, že objekt, ktorý blokuje ostatné predmety je v popredí. Zrková skúsenosť hovorí, že predmet, ktorý je skrytý za iným sa nachádza ďalej.
- Osvetlenie, tieň - Bližšie objekty sú svetlejšie, vzdialené sú tmavšie.
- Relatívny pohyb - Vzdialenejšie objekty sa javia ako pomalšie pohybujúce v porovnaní s objektmi v popredí.



### 3.2.2 Binokulárne klúče

Jedná sa o podnety, ktoré je možné rozoznať pozorovaním pomocou dvoch zdravých ľudských očí.

- Binokulárna disparita - Každé oko vidí odlišný obraz nakoľko sú oči od seba horizontálne vzdialené niekoľko centimetrov. Mozog následne porovnáva tieto obrazy a tak rozdiel retinálnej disparity poskytuje dôležitý kľúč hĺbky.
- Prispôsobenie - Svalové napätie potrebné pre zmenu ohniskovej vzdialenosti šošovky oka s cieľom zamerať sa na určitú hĺbku.
- Konvergencia - Svalové napätie nutné k natočeniu každého oka ku sledovanému bodu.
- Horopter - Teoretickým tvarom horoptera je kruh. Ak je pohľad fixovaný na predmet v určitej vzdialenosti, a tento predmet je viditeľný ostro, tak sú ostro viditeľné aj všetky ostatné predmety v rovnakej vzdialenosti. Predmety pred a za touto vzdialenosťou sú viditeľné dvojmo a skrížene.

### 3.3 Umelé stereoskopické videnie

Ak sú vytvorené podmienky v ktorých je plošný, dvojrozmerný obraz, človekom videný priestorovo, hovorí sa o umelom stereoskopickom videní. Ako prinútiť mozog, aby z plošného obrazu vytvoril priestorový? Mozog funguje jednoducho tak ako väčšina princípov v prírode. Funguje na základe svojich vlastných skúseností, ktoré nazbieral po dobu života človeka v ktorého hlave je umiestnený. Stačí poznať, ktoré faktory sú pri vytváraní priestorového obrazu pre mozog dôležité a následne už nie je problém pomocou jednoduchých trikov mozog oklamať, aby z plošného obrazu vytvoril priestorový zážitok. Jednoduchým pohľadom na túto problematiku je možné tvrdiť, že vlastne nie je treba urobiť nič viac, ako len zabezpečiť, aby každému oku bol doručený obraz, ktorý zodpovedá jeho pohľadu na svet z ľavej alebo pravej očnej bulvy. Tieto obrazy je možné buď vyfotografovať, alebo nakresliť či narysovať, a ak je zachovaná správna rozdielnosť týchto obrazov, zodpovedajúca realite, pretože medzi očami je určitá vzdialenosť, mozog ochotne vyrobí obraz, ktorý je človeku prirodzenejší ako plochý dvojrozmerný obraz, a to obraz priestorový.[5]

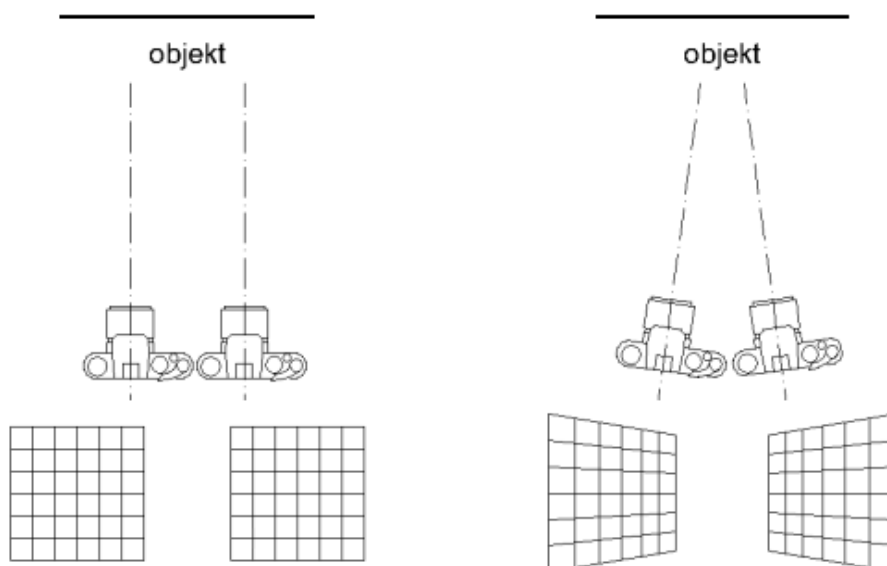
### 3.4 Stereofotografia

3D fotografia je úzko spätá s fyziológiou ľudského zraku, takže na rozdiel od bežnej fotografie, kde sa prehrešky proti pravidlám a technická nedokonalosť prejavia len horším estetickým dojmom, u 3D fotografie môže byť dôsledkom nepohodlie pri pozorovaní či dokonca následné bolesti hlavy a nevoľnosť. 3D fotografia, nie je 3D v pravom slova zmysle, nakoľko plochý obraz nemôžeme vidieť z boku ani zozadu alebo zospodu. Pod pojmom 3D fotografia je myslený pojem

stereogram. Princíp samozrejme funguje na základe stereoskopického zobrazovania. Ten vychádza z poznatkov o optike. Pre správne vnímanie priestoru ľudskými očami je potrebné dodať každému oku mierne odlišný obraz. Do každého oka je potrebné dopraviť tú správnu z dvojice snímok, ako by sa na svet pozeral dvoma šošovkami fotoaparátu. [6]

### 3.4.1 Vytvorenie stereoskopického páru

Pri tvorbe kvalitného stereopáru je nutné dodržať pravidlo, že obe snímky aj v individuálnom prezeraní produkujú prijateľný obraz s vizuálnym kontextom. Celý záber by mal byť v hĺbke ostrosti, tzn. všetko na snímke by malo byť ostré. Použitie teleobjektívu alebo výrazne širokouhlého objektívu nie je chyba, ale pôsobí neprirodzene (vhodný je rozsah 35 až 50mm v prepočte na kinofilm, tzn. zorný uhol podobný ľudskému zraku). Aj nepostrehnuteľne jemné rozdiely medzi pravou a ľavou snímku vytvárajú priestorovú informáciu. Fotografia je teda prirodzenejšie viac priestorová čím má väčšie rozlíšenie. Musí byť vylúčená vzájomná rotácia snímok a obrazy rovnakých bodov musia byť na ľavej aj pravej fotografii v rovnakej vertikálnej výške. Snímané objekty nesmú byť v pohybe. Výnimkou je snímanie s použitím dvoch objektívov, no aj v tom prípade je problematické vyfotografovať pohybujúce sa objekty s dostatočnou ostrosťou. Pokiaľ je to možné, fotoaparáty majú vždy smerovať rovnobežne. V prípade fotografovania s jedným fotoaparátom platí samozrejme pravidlo, že prvá a druhá snímka musia zostať na rovnakej priamke s rovnakým naklonením vo všetkých smeroch, ako na obrázku 1 vľavo. Ľavá a pravá snímka majú zhodné perspektívne skreslenie. Ak nie sú fotoaparáty pri snímaní rovnobežné, na jednotlivých snímkach sa bude líšiť vertikálna poloha jednotlivých bodov v obraze, ako na obrázku 1 vpravo čo vyústí v rozdielne perspektívne skreslenie. Tento jav je v 3D fotografii neprípustný, nakoľko je nutné aby sa body na jednotlivých snímkach prekrývali.



Obr. 1: Vľavo správna pozícia fotoaprátu, vpravo nesprávna. Prevzaté a upravené z [23]

Je treba mať na pamäti, že niektoré pravidlá používané pri klasickej fotografii neprinášajú správny efekt, alebo len v obmedzenej miere, pretože do výslednej snímky pridávame tretí rozmer (skôr pseudo rozmer, nakoľko fotografia len klame jeden z procesov ktorým mozog vyhodnocuje vzdialenosť). Pokiaľ sa pri fotografovaní, zapracovaní a prezentácii pravidla rešpektujú, dá sa týmto dôsledkom účinne predchádzať. Pre správne prevedenie stereoskopickkej snímky je nutné stanoviť rozsah priestoru, ktorý je prípustné zachytiť. Podmienky pri ktorých pozorujeme 3D fotografiu a skutočný svet nie sú úplne totožné. Ako príklad môže byť uvedený konflikt spôsobený tým, že zbíhavý pohyb očí (konvergencia) a ich zaostrenie (akomodácia) sú na seba viazané. Keď je pozorovaný vzdialený predmet v reálnej scéne, oči sú nasmerované rovnobežne a majú zaostrené na diaľku. Pokiaľ je pozorovaný blízky predmet, zbíhajú sa a zaostrené majú na blízko. Pri pozorovaní 3D fotografie sú však oči nútené byť stále zaostrené na rovinu fotografie (diapozitívu, monitoru, projekčného plátna), zatiaľ čo obraz obsahuje body s rôznym vodorovným posunom. Čím viac je na fotografii zachyteného priestoru, tým viac nútime oči k namáhavým a neprirodzeným pohybom. Pokiaľ prekročíme prijateľnú mieru "priestorovosti" snímky, teda pokiaľ záber obsahuje zároveň body príliš vzdialené a príliš blízke, vedie to k neprirodzenému namáhaniu očí, v krajnom prípade vôbec nedôjde k spojeniu obrazov. [6]

### 3.5 Určenie vhodnej miery priestorovosti snímky

Prijateľné množstvo priestoru nie je hodnota, ktorá by sa dala exaktne stanoviť. Je individuálna, závisí na vzdialenosti očí pozorovateľa a na skúsenostiach. Toleranciu očí je možné do značnej miery trénovať, preto si skúsený pozorovateľ spojí aj nevhodné obrazy. Ovplynená je aj spôsobom pozorovania teda zorným uhlom. Súvisí aj s kompozíciou snímky a ďalšími veľmi ťažko vyčísliteľnými a zovšeobecniteľnými javmi. Preto je nutné vychádzať z empiricky stanovených a všeobecne uznávaných hodnôt. Snímka s prehnaným množstvom priestoru sa stáva nevhodnou až nepoužiteľnou, avšak striedma priestorovosť ničomu neuškodí. Časté uvažovanie, hlavne u začiatočníka, je také, že snímku je vhodné urobiť čo najpriestorovejšiu - teda zvoliť maximálnu možnú základňu, akú nám umožnia pravidlá výpočtu deviácie. Je však nutné si uvedomiť, že je to za cenu menej prirodzeného dojmu z priestoru. Pri prehliadaní fotografií vyfotografovaných s ľahko zväčšenou bázou nám začne pripadať zmenšenie pozorovaných objektov (hlavne postáv) nepríjemné. Avšak niekedy aj výrazne hyperstereoskopická snímka krajiny, alebo makro môžu byť úžasným zážitkom, ktorý zobrazí reálny svet ešte strhujúcejším spôsobom než je v skutočnosti. [7]

Vnem pozorovateľa pozorujúceho 3D obraz je možné rozdeliť následovne:

- **Orthostereo** označuje snímku, ktorej zobrazený priestor verne napodobňuje skutočný pohľad ľudským zrakom. Nastáva pokiaľ sú splnené nasledujúce predpoklady: Snímka je vyfotografovaná so stereoskopickou základňou podobnou vzdialenosti očí, teda približne v rozmedzí medzi 55 až 70 mm. Snímka je vyfotografovaná s prirodzeným zorným uhlom (objektív 35 až 50 mm v prepočte na kinofilm). Snímka je pozorovaná pod podobným

zorným uhlom, ako bola vyfotografovaná. To je zaručené u správne skonštruovaných šošovkových alebo zrkadlových stereoskopov, ale komplikovanejšie je to s projekciou alebo anaglyfom. [7]

- **Hyperstereo** Týmto termínom sa rozumejú snímky, ktoré sú viac priestorové, než jak vidíme skutočnosť. Typicky toho docielime zväčšením stereoskopickej základne nad bežnú vzdialenosť ľudských očí. Teda viac, ako 7cm, ale môžu to byť aj desiatky či stovky metrov pri fotení vzdialenej krajiny, alebo ešte niekoľko krát viac pri hviezdárskych fotografiách. Základňu je možné zväčšiť samozrejme len ak je najbližší objekt na fotografii dostatočne vzdialený, aby nebola prekročená prijateľná deviácia. Priamo úmerne so zväčšením základne rastie aj vzdialenosť, do ktorej vidíme priestorovo. V reálnom svete vidíme najlepšie plasticky objekty niekoľko metrov vzdialené a jednotlivé vrstvy priestoru rozlišujeme až do stoviek metrov. Rovnako je to u orthostereoskopickej snímky s dokonalým rozlíšením. U hypersterea s desaťnásobnou základňou (60cm) uvidíme dobre plasticky desiatky metrov a priestor rozoznáme do niekoľkých kilometrov. Všetky objekty na fotografií vyfotografované so zväčšenou bázou sa pozorovateľovi javia menšie, ako v skutočnosti. Ako by sa stal človek obrom, zväčšeným v rovnakom pomere, ako je zväčšená vzdialenosť jeho očí. Ak je zvolená základňa 12cm, teda približne dvojnásobok vzdialenosti očí, úmerne k tomu sa napr. osoby zdajú byť polovičné. Krajina vyfotografovaná so základňou 6m, teda stonásobkom vzdialenosti očí, vypadá ako dokonalý model v mierke 1:100. [7]
- **Hypostereo** Hypostereo je opačný prípad - snímka vyfotografovaná so základňou menšou, ako vzdialenosť očí. Často je základňa zmenšená, keď nie je možné zaistiť dostatočný odstup od najbližšieho bodu. Patrí sem aj makrofotografia so základňou v radoch milimetrov či mikroskopické snímky. Objekty na takto vytvorenej snímke sa javia analogicky väčšie, ako sú v skutočnosti. Napr. na makrofotografii môže veľkosť chrobáka, alebo mikroskopických húb zodpovedať polmetrovému objektu. [7]

Pokiaľ nepracujeme v ateliéri alebo za iných výnimočne dobrých podmienok, je samozrejme nepraktické až neprevediteľné merať vzdialenosti najbližších predmetov v scéne a počítať potrebné hodnoty pre každú snímku zvlášť. Pre bežné fotografovanie postačí zapamätať si základné údaje. Názorný príklad by mohol byť pri fotografovaní s objektívom 50 mm a prirodzenou základňou (cca 60 mm), žiadny objekt nesmie byť bližšie, než cca 2.5m. [7]

### 3.6 Spôsobý zobrazenia 3D snímky

3D alebo 3-D je skratka výrazu „trojdimenzionálny“, „trojrozmerný“ a označuje svet, ktorý je možné popísať tromi rozmermi. Označuje často aj techniky používané pre zobrazenie a prehliadanie zdanlivo trojrozmerných objektov na plochom dvojrozmernom 2D médiu (papier, filmové plátno, LCD obrazovke apod.).



### 3.6.1 Techniky bez pomôcok

Sledovať každým okom inú snímku nie je bez pomôcok ľahké pretože oči majú na seba viazanú konvergenciu a akomodáciu. To znamená, že oči sa buď dívajú rovnobežne a majú pri tom zaostrené na nekonečno, alebo sa zbiehajú a majú zaostrené na blízko. Izolovanie týchto dvoch pohybov vyžaduje cvičenie, a nie každému sa to podarí. Tento spôsob pozorovania je vhodný len na príležitostné použitie, nakoľko metódy vyžadujú neprirodzené pohyby očí, čo spôsobuje následný diskomfort.

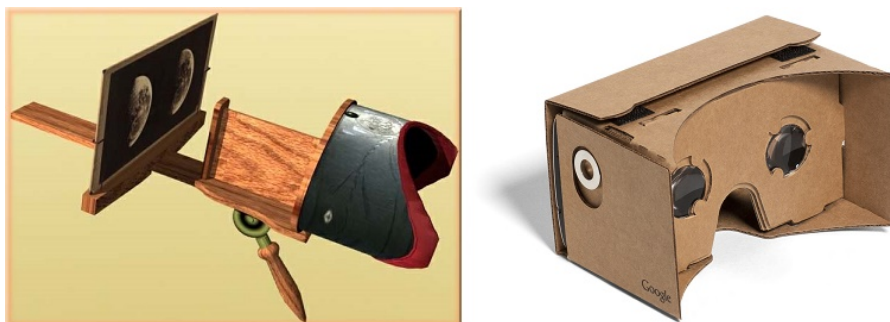
**Rovnobežne (parallel)** Oči pozorovateľa sú nasmerované do diaľky ako by sa pozeral von oknom. Oči by mali byť uvoľnené z dvoch snímok sa vytvoria tri. V tej chvíli oči preostria na rovinu obrazu a z strednej snímky vznikne priestorový obraz. Problémom by bolo, ak by šírka snímok bola väčšia ako vzdialenosť očí pozorovateľa. Ak by nastal tento prípad museli by byť snímky pozorované z väčšej diaľky alebo by pohľad očí musel byť rozbiehavý, čo sa nazýva divergencia. Je to možné, ale nie jednoduché a vyžaduje to cvik.

**Krížom (cross-eyed)** Pre tento účel sa ľavá fotografia umiestni vpravo a pravá vľavo a oči škúlia krížom na správnu snímku. Pre väčšinu ľudí je tento spôsob jednoduchší a snímky môžu byť väčšie, avšak pre skúsenejšieho pozorovateľa je príjemnejší prvý uvedený spôsob.

### 3.6.2 Zobrazenia pomocou technológií

3D snímky je možné pozorovať pomocou veľmi jednoduchých pomôcok, ako je stereoskop, alebo anaglyf ale taktiež pomocou zložitých technológií, ako sú televízory s funkciou sledovania očí.

**Sledovanie pomocou Stereoskopu** Stereoskop je optické zariadenie 2, ktoré nasmeruje pohľad každého oka na správny obraz pomocou šošoviek, hranolov či zrkadiel. Buď teda oči hľadia rovnobežne (akoby do nekonečna), ale okuliare preostria na rovinu obrazu, alebo sa pohľady zbiehajú na blízky bod a pohľad je hranolom či zrkadlom zalomený správnym smerom. Je možná aj kombinácia oboch princípov. Rovnako ako u fotografického objektívu, aj tu je dôležitá kvalita použitej optiky - pôžitok pri pozorovaní fotografie lacnou plastovou šošovkou nie je taký, ako s kvalitným vybavením. Najbežnejšími druhmi stereoskopov sú kukátka na diapozitívy, do nich sa zasúvajú kinofilmové alebo stredno formátové rámčeky alebo kotúčky, kukátka pre pozorovanie papierových dvojíc (napr. historické stereoskopy alebo súčasné skladacie papierové pohľadnice) alebo v ruke držané okuliare pre použitie na monitore či v knihách. Pomocou jedného zrkadla je možné veľmi kvalitne pozorovať obrazy na dvoch monitoroch, či veľké papierové dvojice. Správne skonštruovaný a kvalitný stereoskop, s obrázkami o veľkom rozlíšení, je najdokonalejším spôsobom prehliadania 3D fotografií. Prehliadačka s stredno formátovým diapozitívom zrejme ešte dlho zostane bezkonkurenčnou technológiou, čo sa kvality vnemu týka. Nevýhoda je očividná - obraz vidí len jeden divák, nútený pozeráť do viac či menej zložitého prístroja. [16]



Obr. 2: Stereoskop, Google Cardboard. Prevzaté z [16] [27]

**Polarizované premietanie** Svetlo je tvorené elektromagnetickými vlnami, ktoré za bežných okolností kmitajú vo všetkých smeroch. Pokiaľ je svetlo vysielané pre jedno oko s vlnami kmitajúcimi len vodorovne a pre druhé len zvislo - a sú použité polarizačné okuliare, ktoré ku každému oku prepustia žiadané vlny, pozorovateľ uvidí 3D obraz. Premieta sa špeciálnym projektorom s dvoma objektívmi na plátno, ktoré nerozbíja polarizáciu. Túto technológiu dnes používa napríklad kino Imax alebo moderné 3D TV. [18]

**Aktívne okuliare** Ľavý a pravý obraz sa s vysokou frekvenciou striedajú na plátne či monitore, a divák má okuliare u ktorých sa striedavo zakrýva pravé a ľavé oko. Okuliare sú s počítačom synchronizované infračerveným spojením. [17]

**Auto-stereoskopické technológie** Jedná sa o vytvorenie obrazu, ktorý vidíme priestorovo prirodzeným pohľadom, bez akýchkoľvek pomôcok. Spoločným znakom všetkých auto-stereoskopických monitorov je špeciálna maska umiestnená pred LCD displejom. Je vybavená optickými hranolmi, ktoré vychylujú rôzne stĺpce pixelov do rôznych smerov. Na obrazovke monitoru sú podľa parametrov optickej masky vtesnané obrazy pre pravé aj ľavé oko. Maska spôsobí to, že sú obrazy pre jednotlivé oči viditeľné len z určitých smerov. Pokiaľ sa teda divák postaví do vhodnej pozície, pričom pre každého človeka môže byť iná, a to podľa vzdialenosti očí, uvidí ľavým okom len ľavý obraz a pravým pravý. Nevýhodou monitoru je to, že je kvalita každého obrazu len polovičná, než je natívne rozlíšenie. [17]

**Auto-stereoskopické monitory so sledovaním očí (Eye-tracking)** Sledovanie 3D obrazu na auto-stereoskopickom monitore nie je príliš pohodlné, divák si musí nájsť pozíciu pred monitorom, kde 3D funguje a on ho vidí dobre. Tento neduh sa snaží odstrániť systém pre sledovanie očí. Takéto monitory sú vybavené kamerovým systémom, ktorý sleduje oči diváka a fyzicky posúva optickú masku pred monitorom tak, aby sa menili pozorovacie zóny v závislosti na pozícii diváka pred monitorom. [17]

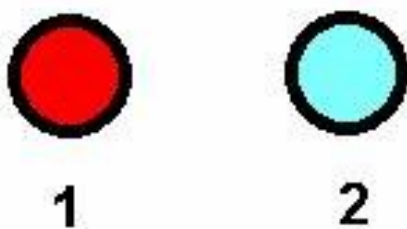
**Wiggle stereoscopy** Wiggle je príkladom 3D vizualizačnej technológie v ktorej ľavý a pravý obraz stereoskopického páru sú animované. Táto technika sa tiež nazýva wobble 3-D alebo Piku-

Piku. Princípom vnímania hĺbky obrazu pri tejto vizualizačnej technike je takisto posun fotoaparátu a neustála rýchla animovaná výmena ľavej snímky za pravú. Táto technika sa od ostatných odlišuje tým, že ako jediná premieta snímky do oboch očí pozorovateľa. Animácie môžu byť zobrazené len na displeji a to vo formáte GIF, flash animácia alebo JavaScriptový program. [14]

**Anaglyf** Najmenej náročnou ale zároveň bohužiaľ aj najmenej kvalitnou metódou stereoskopie je anaglyf. Pozorovateľ má okuliare s rôznymi farebnými filtrami - najčastejšie červeným a modrozeleným (cyan), ale môže to byť akákoľvek dvojica komplementárnych farieb. Z jednej snímky je vynechaná červená farebná zložka (R), z druhej ostatné dve zložky (G+B). Následne sú obrazy opäť zjednotené do jedného, sčítaním hodnôt RGB. Anaglyfy fungujú na monitore, vytlačené na papieri aj premietnuté na plátne - jedná sa teda o metódu jednoduchú, dostupnú, a veľmi lacnú. Avšak separácia obrazov nie je dokonalá, a tak kvalitu zásadným spôsobom znižujú viac či menej viditeľné "duchy". Obrovským nedostatkom je skreslenie či úplná strata farieb.

Test okuliarov:

Správnosť a presnosť red-cyan okuliarov je možné overiť pomocou nasledujúcich dvoch farebných bodov. 3



Obr. 3: Anaglyf test

- Červené sklo je nasadené na ľavom oku a modré na pravom.
- Po zatvorení ľavého oka : kruh 1 má čiernu výplň, kruh 2 je bez výplne.
- Po zatvorení pravého oka : kruh 1 je bez výplne, kruh 2 má čiernu výplň

Okuliare neprešli testom, ak výplň kruhu nie je úplne tmavá a kruh bez výplne nie je rovnakej farby ako okolie. Ak nastali malé farebné rozdiely, tak okuliare pravdepodobne nie sú kvalitné, ale ešte je šanca, pomocou nich pozorovať priestorovosť anaglyfu, avšak s neprirodzenými farbami. Pozorovanie anaglyfu ma lepšiu priestorovosť v porovnaní so zrkadlovým stereoskopom, avšak snímky trpia farebnou deformáciou. Zrkadlové stereografie si jednoducho pozriete priložením zrkadielka k obrazovke. Priestorovosť je horšia, treba sa nakloniť blízko k monitoru a fotky musia byť tak malé, aby sa vošli hneď dve vedľa seba na obrazovku. Výhodou anaglyfu v porovnaní so zrkadlovým stereoskopom je jeho lepšia priestorovosť. Stereoskop však disponuje prirodzenejším zobrazením farieb a je použiteľný aj pri väčších rozmeroch snímky nakoľko pri

pozorovaní zrkadlovým stereografom je nutné aby boli snímky tak malé, aby sa vošli dve vedľa seba. [11]

**Ďalšie riešenia** 3D prilby alebo 3D okuliare s dvomi farebnými displejmi (jedným pre každé oko), dvojica monitorov oddelená prepážkou alebo dvojica naklonených zrkadiel. Dokonalé riešenie - teda lacné, ľahko prenosné, neobmedzujúce, neskresľujúce a ponúkajúce pohľad z rôznych uhlov zatiaľ neexistuje. Vývoj ale ide ďalej a je možné, že jedného dňa budú 3D technológie štandardným vybavením.

### 3.7 Snímanie stereoskopického páru

Pre vytvorenie 3D snímky existuje niekoľko postupov pričom ani jeden z nich nie je úplne jednoduchý, avšak pri dodržiavaní pravidiel sa môže komukoľvek podať vytvoriť dokonalý stereoskopický pár.

#### 3.7.1 Vytvorenie stereoskopických snímok v rovnakom okamžiku pomocou dvoch objektívov 3D fotoaparátu.

3D fotoaparáty sú zostavené tak, aby čo najviac napodobňovali ľudské videnie, a to jak uhlom záberu tak aj roztečou objektívu. 3D fotoaparát má dva rovnaké objektívy, synchronizovaný uzáver, clonu aj ostrenie. Princíp spočíva v zachytení dvoch plošných snímok, ktoré sú vďaka konštrukcií zariadenia snímané z rôznych uhlov a predmet sa následne vo výsledku javí, ako umiestnený v priestore.[13]

#### 3.7.2 Synchronizácia dvoch fotoaparátov

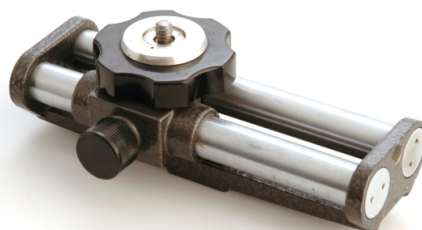
Je možné použitie dvoch obyčajných fotoaparátov alebo smartfónov a mechanicky alebo elektricky ich synchronizovať. Synchronizácia je väčšinou menej presná ako u stereofotoaparátu. Prináša však možnosť ľubovoľnej stereoskopickkej báze (rozteč objektívov). Pre tieto potreby existuje Android aplikácia Hyper3DPhone, ktorá umožňuje cez bluetooth synchronizovať dve Android zariadenia a snímať stereosnímky.[13]

#### 3.7.3 Posunutie fotoaparátu

Nepohyblivé objekty je možné vyfotografovať aj postupne s jedným fotoaparátom, tak, že sa medzi ľavou a pravou snímku fotoaparát posunie o stereoskopickú bázu. Nevýhodou je, že sa medzi oboma expozíciami nesmie na fotografií nič pohnúť. Mimo živých bytostí a dopravných prostriedkov dokáže snímku pokaziť aj pohyb lístia vo vetre, pohyb vody, mrakov alebo ich tieňov. Táto metóda je teda striktne obmedzená na fotenie statických scén za dobrých podmienok. Ďalšou nevýhodou je, že nie je možné použitie blesku umiestneného na fotoaparáte, použitie blesku je možné jedine mimo fotoaparátu, tak aby sa nepohol s posunom fotoaparátu. Dôvod je zrejímavý, došlo by k porušeniu podmienky, že každá snímka zo stereoskopického páru musí byť



zosnímaná v rovnakých svetelných podmienkach, pričom posun blesku radikálne mení nasvietenie scény. Pri tejto metóde snímania je veľmi spoľahlivé použitie statívu spolu s posuvnou lištou nazývanou stereosánky 4, ktorá umožní fotoaparát presunúť o presne danú vzdialenosť, bez odchýlky či rotácie. Posun fotoaparátu, je možné zabezpečiť aj z ruky, ale je ťažké odhadnúť presnú dĺžku báze a vyvarovať sa odchýlkam. Dĺžku prirodzenej základne je možné dosiahnuť prenesením váhy z ľavej nohy na pravú, alebo pomocou jednoduchého triku. Jedno oko je zatvorené a následne druhé. Je treba zapamätať si obe polohy jednoznačného bodu z popredia voči pozadiu. Následne je fotoaparát priložený k oku a je zopakovaný rovnaký postup. Táto metóda zároveň pomáha vyvarovať sa zvislej odchýlky tzn. nakloneniu. Fotenie 3D snímok z ruky nie je jednoduché a chce to prax, avšak je veľmi praktické a rýchle, bez nutného nastavovania, alebo nosenia statívov. Ako veľmi užitočná pomôcka by bolo navádzanie pomocou senzorov, čo už v dnešnej dobe plnej smartphonov prešpikovaných senzormi, nie je nerealizovateľné. Výhodou snímania s posunom fotoaparátu je, že je možné zvoliť ľubovoľne veľkú bázu. [13]



Obr. 4: Stereo sánky prevzaté z [13]

### 3.8 Formáty uloženia 3D snímky

Dôležitou súčasťou vytvorenia snímky je výber formátu uloženia. K ukladaniu fotografií existuje rada celá rada formátov a z nich sú najpoužívanejšie GIF, JPG (JPEG), TIF (TIFF), BMP a PNG. Každý z týchto formátov má svoje prednosti ale aj nedostatky. Formáty GIF a JPG sú obľúbené vďaka svojej úspore miesta na disku, a preto sú často využívané pri tvorbe webu. Obidva tieto formáty sú komprimované a JPG navyše využíva takzvanú stratovú kompresiu. To znamená, že sa niektoré informácie pri ukladaní nenávratne stratia a tiež sa napríklad rozmazú drobné detaily. To je nepríjemné u 3D obrazu, pretože tu ostré detaily pomáhajú získať dobrý priestorový vnem. Kvalita kompresie je nastaviteľná, avšak čím lepší obraz, tým viac miesta zaberie na disku. GIF používa bezstratovú kompresiu, avšak štandardne zvládne uložiť maximálne 256 farieb, čo je na 3D fotografie často veľmi málo. Formát TIF ponúka bezstratovú kompresiu a veľa farieb, na disku však jeho obrázky zabierajú viac miesta. BMP sa kompresiou príliš nezťažuje, zato však pokúša dobrú kompatibilitu a zachovanie kvality obrazu aj farieb. Ide o ideálny formát pre tvorbu vlastných 3D obrazov. Pokiaľ je obraz uložený v BMP, môže sa ďalej

prevádzať do formátov TIF, GIF napríklad pre vystavenie na internet, avšak obrázok v plnej kvalite ostáva zachovaný vo formáte BMP. [15]

## 4 Popis metód automatického spojenia stereoskopického páru

Ak by hardware parametre mobilného zariadenia boli dostatočne vysoké a robustnosť mobilnej aplikácie by nebola problémom, bolo by možné v Android aplikácií implementovať automatickú metódu spojenia stereoskopického páru, na základe automatickej detekcie zhodných bodov. Pre tento účel by sa mohli hodiť algoritmy typu SIFT alebo SURF.

- **SURF** je metóda, ktorá dokáže popísať obrázok pomocou deskriptorov. Jedná sa o novšiu podobu metódy SIFT. Popis pomocou deskriptorov vygenerovaných metódou SURF je invariantný voči rotácií a vzdialenosti kamery od popisovaného objektu. Algoritmus SURF sa využíva v mnohých aplikáciach počítačového videnia. Je používaný napr. pre rekonštrukciu 2D a 3D scén, klasifikáciu obrázkov a predovšetkým pre rýchly popis obsahu obrázku. Mieru podobnosti dvoch obrázkov je možné merať napr. pomocou Euklidovskej vzdialenosti. Priebeh metódy SURF je možné rozdeliť na dve fázy. V prvej fáze sú hľadané kľúčové body obrázkov, ktorými môžu byť rohy, škvrny alebo T-spoje. Druhou fázou je výpočet deskriptorov z okolia kľúčového bodu. [24]
- **SIFT** Je škálovo invariantná transformácia črt. Týmto prístupom je obrázok transformovaný do množiny vektorov lokálnych črt, pričom každý z nich je invariantný voči rotácií, skresleniu a posunutiu. Deskriptory typu SIFT sú získavané z detekcie extrémov škálového priestoru za pomoci rozdielu dvoch Gaussových funkcií, lokalizovaných kľúčových bodov, priradenej orientácie a zo zostaveného deskriptora kľúčového bodu. [25]

## 5 Porovnanie existujúcich aplikácií z kategórie stereoskopického snímania

Logickým krokom pred samotným vývojom, či návrhom novej aplikácie je prieskum existujúcich aplikácií. V dnešnej dobre preplnenej technológiou, je veľmi pravdepodobné, že keď človek dostane technologicky nový nápad, tak niekto pred ním už nielenže dostal nápad, ale ho aj zrealizoval. Bolo by veľmi nerozumné pokúšať sa preraziť na trh mysliac si, že sa ešte nikto nepokúsil o niečo podobné, bez akéhokoľvek prieskumu trhu.

Prvým krokom prieskumu dostupných alternatív Android aplikácií pre vytváranie stereoskopických snímok bolo jednoduché zadávanie kľúčových slov do vyhľadávača obchodu Play store.

Bolo zistené, že záujem používateľov o aplikácie so zameraním na vytvorenie 3D fotografií je relatívne veľký, avšak tomu nezodpovedá počet použiteľných aplikácií nachádzajúcich sa v obchode Play Store. Počet stiahnutí aplikácií nezodpovedá počtu užívateľov, ktorí túto aplikáciu naozaj používajú. Je pravdepodobné, že keď už sa raz niekto zaujíma o tak špecifickú oblasť fotografovania, ako je vytváranie 3D snímok a narazí na aplikáciu, ktorá spĺňa jeho predstavy, bude túto aplikáciu využívať až do doby, než sa na trhu objaví lepšia alternatíva.

### 5.1 Požadované vlastnosti a chovanie aplikácií

Aplikácií v kategórii fotografia s označením 3D je v obchode Play store niekoľko, avšak nutnou podmienkou aby aplikácia bola zaradená do nasledujúceho prieskumu bolo, aby aplikácia umožnila nasnímanie stereoskopického páru a následne ponúkala možnosť uloženia v niektorej zo známych 3D vizualizačných techník a zároveň musí byť ponúkaná bezplatne. Tieto podmienky do značnej miery preriedili prvotný výber aplikácií, pretože označenie 3D vo fotografií môže označovať nie len stereofotografické zobrazovanie, ale aj takzvané 3D modelovanie. 3D modelovanie je zobrazenie predmetu v priestore s možnosťou otáčania priestorom a tým pádom nahliadania na predmet z 360 perspektívy. Aplikácie s týmto zameraním nemohli byť zaradené do prieskumu, nakoľko sa jedná o diametrálne odlišnú funkcionálnosť.

### 5.2 Popis a testovanie aplikácií

Po nainštalovaní a otestovaní vybraných aplikácií na základe popisu a hodnotenia užívateľov vyplynulo, že požadované vlastnosti a chovanie spĺňujú len nasledujúce aplikácie umiestnené v tabuľke 1

#### 5.2.1 3D Steroid

Jednoznačne najlepšie spracovaná a najúspešnejšia z dostupných aplikácií. Aplikácia je v podstate vybavená všetkou 3D funkcionálnosťou do tej doby poznanou, okrem snímania fotografií pomocou prednej kamery. Dokonca používa senzori zariadenia a pomáha používateľovi udržať horizontálnu rovinu. Pri fotografovaní sa na spodnej strane obrazovky nachádzajú dve

Aplikácia > Funkcia	3D Steroid	Make it 3D	3D Camera	Easy 3D
Aspoň dve rôzne vizualizačné techniky	Áno	Áno	Áno	Nie
Anaglyf	Áno	Áno	Nie	Áno
Senzorické navádzanie	Áno	Nie	Nie	Nie
Korekcia naklonenia	Áno	Áno	Áno	Nie
Korekcia rotácie	Áno	Áno	Nie	Nie
Manuálne spojenie snímok	Áno	Áno	Áno	Nie
Poloautomatické spojenie snímok	Nie	Nie	Nie	Nie
Fotografovanie prednou kamerou	Nie	Áno	Nie	Nie
Transparentné zobrazenie prvej fotografie pri snímaní druhej fotografie	Áno	Nie	Áno	Nie
Reálne zobrazenie pomeru strán prvej fotografie pri snímaní druhej fotografie	Nie	-	Nie	-
Zobrazuje reklamy	Nie	Áno	Áno	Áno
Najvyššie rozlíšenie snímok	1080/720 778KP	1024/768 786KP	Neukladá snímky	1024/768 786KP

Tabuľka 1: Porovnanie existujúcich aplikácií

úsečky. Jedna(bielá) je vždy paralelná so zemou vďaka komunikácií so senzormi zariadenia a druhá(červená) je paralelná so spodnou a hornou stranou obrazovky zariadenia. Vo chvíli, kedy sa tieto čiary prekrývajú, je jasné, že nastáva ideálny okamžik pre vytvorenie snímky, pretože je zariadenie paralelne so zemou a teda objekty na snímke nebudú v žiadnej inej pozícii ako vo vodorovnej. 3D Steroid fotografie ukladá v rozlíšení 1080/720 čo je najvyššie rozlíšenie z aplikácií z prieskumu. Pri spustení neobťažuje používateľa reklamami a jej používateľské rozhranie je veľmi prehľadné. Obchod play store uvádza, že počet inštalácií tejto aplikácie sa pohybuje v rozmedzí sto tisíc - pol milióna pričom aplikáciu hodnotilo celkovo viac ako 1500 používateľov a jej výsledné hodnotenie je 4.1 na škále od 1 do 5 hviezdíček.

### 5.2.2 Make It 3D Free - 3D kamera

Aplikácia Make It 3D Free je výnimočná tým, že ako jediná z dostupných aplikácií umožňuje fotografovanie stereoskopického páru pomocou predného fotoaparátu, avšak používateľom nijako nepomáha pri vytváraní druhej snímky, takže je veľmi náročné a skoro až náhodné, vytvorenie kvalitného stereoskopického páru. Pri fotení je používateľ nútený spoliehať sa na presnosť vlastnej ruky pričom je zároveň rozptyľovaný neustále blikajúcimi reklamami na vrchnej strane displeja.

Počet stiahnutí je síce väčší ako u aplikácie 3D Steroid, avšak hodnotenie sa zastavilo na hodnote 3.8 hviezdíčky z 5. To všetko radí aplikáciu Make it 3D na druhé miesto tohto výberu.

### **5.2.3 3D Camera**

Z popisu dostupných funkcií tejto aplikácie z tabuľky 1 jasne vyplýva, že neprináša nič nové v porovnaní s predchádzajúcimi aplikáciami. Ba dokonca umožňuje prehliadanie snímok len vrámci aplikácie, pričom neumožňuje exportovanie snímky na kartu SD, čo je dosť zásadný nedostatok. Počet stiahnutí podľa obchodu play store prekročil jeden milión, avšak hodnotenie 3.6 hviezdíčky od viac ako 22 tisíc používateľov jasne hovorí, že táto aplikácia patrí medzi podpriemer vo svojej kategórii.

### **5.2.4 Easy 3D Camera FREE, Easy Wiggle Image Creator FREE**

Autor týchto aplikácií si povedal, že chce aby boli čo najjednoduchšie, a tak sa rozhodol, že vyrobí jednu aplikáciu, ktorá zo stereoskopického páru vytvorí anaglyf a druhú aplikáciu, ktorá zo stereoskopického páru vyrobí wiggle vizualizačný formát. Ťažko povedať prečo autor týchto aplikácií vyrobil dve aplikácie s rovnakým dizajnom a celkovou funkcionalitou, pričom ich jediným vzájomným rozdielom je, že každá z nich vytvára zo stereoskopického páru iný 3D vizualizačný formát. Podľa autorovho popisu v obchode play store sa pokúsil aplikáciu zjednodušiť najviac ako to len šlo, pretože dostával sťažnosti, že používatelia sa v jeho aplikácii nevyznajú.

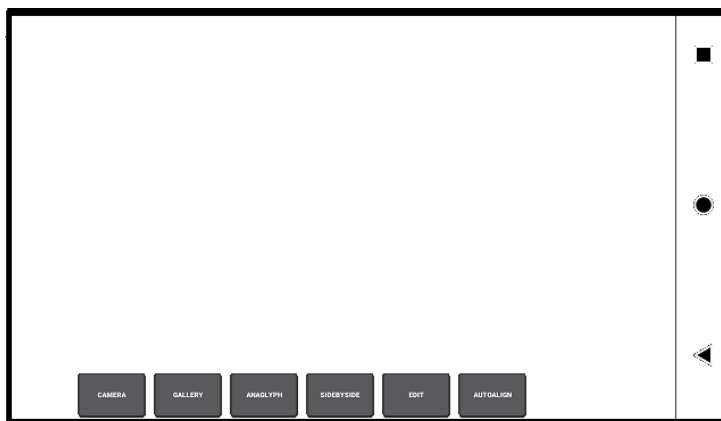


## 6 Návrh aplikácie Camera3D

Na základe zadania, popisu problematiky a existujúcich riešení vznikol súbor funkčných a nefunkčných požiadavkov. Vývoj používateľského rozhrania bol vykonaný pomocou iteratívneho vývoja. Základom návrhu aplikácie Camera3D bol prieskum trhu mobilných Android aplikácií súvisiacich s 3D fotografiou. Bolo nutné analyzovať, ktoré funkcie existujúcich aplikácií by sa dali vylepšiť a zväžiť, či by sa dala vymyslieť funkcia, vďaka ktorej by Camera3D získala náskok pred ostatnými svojho druhu. Ideou návrhu aplikácie Camera3D bolo, aby vznikla aplikácia, ktorá si vezme z každej existujúcej aplikácie, z kategórie stereoskopia, všetky zaujímavé funkcie, ktoré vylepší, a ako bonus pridá funkcie, ktoré nie sú obsiahnuté v žiadnej z existujúcich voľne dostupných verzií aplikácií.

### 6.1 Iteratívny návrh používateľského rozhrania

Spolu s návrhom používateľského rozhrania snímania fotografií a indikácie vhodného okamžiku 7,8 bolo treba navrhnuť aj rozhranie pre používanie hlavných funkcií aplikácie. Pre tento účel bol použitý nástroj Pencil a návrh v konečnej iterácii vypadal nasledovne 5



Obr. 5: Návrh UI

### 6.2 Funkčné požiadavky

Cieľom pri tvorbe aplikácie bolo ponúknuť širokú škálu funkcií v snahe zaujať čo najväčšie spektrum používateľov, zároveň však predstaviť funkcie, na ktoré bolo nahliadnuté z inej, odlišnej a téme prospešnejšej perspektívy. Camera3D obsahuje všetky funkcie zistené v prieskume z tab.1 a niektoré funkcie výrazne vylepšené. Pozornosť bola sústredená na novú funkciu indikácie vhodného okamžiku snímania pri vytváraní stereoskopického páru, pretože je vždy lepšie vyfotografovať takmer dokonalú dvojicu snímok, ako vyfotografovať nie príliš dobré snímky a snažiť sa ich neskôr zložitými algoritmami upraviť do prijateľnej formy. Nasleduje podrobný popis na-

vrhnutých funkčných požiadavkov aplikácie Camera3D, ktoré boli vytvorené v zmysle splnenia jednotlivých bodov zadania diplomovej práce.

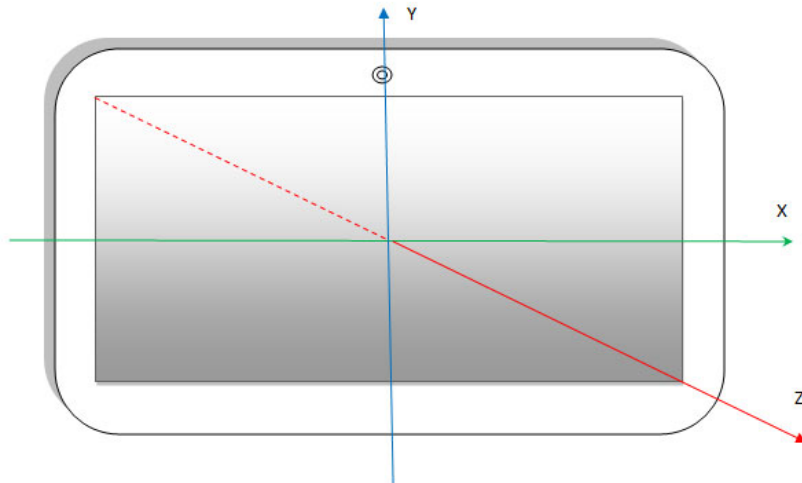
### 6.2.1 Prepínanie medzi prednou a zadnou kamerou

V aplikácii sa nachádza funkcia dostupná pri fotografovaní, na prepnutie medzi prednou a zadnou kamerou, čo logicky používateľovi umožní urobiť si vlastnú 3D selfie a zdieľať ju na niektorej z, v dnešnej dobe tak populárnych, sociálnych sieťach.

### 6.2.2 Indikácia vhodného okamžiku snímania - senzorické navádzanie

Ako bolo skôr uvedené, tak na funkcionality senzorického navádzania pri fotografovaní v aplikácii Camera3D bola sústredená veľká pozornosť, nakoľko samotné procesy predchádzajúce nechceným nakloneniam a rotáciám sú dôležitejšie, ako neskoršie úpravy týchto chýb, vzniknutých pri fotografovaní. Ak boli obrázky chybné vytvorené, len zriedka sa podarí upraviť ich do použiteľnej formy pre 3D zobrazenie. Za vzor senzorického navádzania v aplikácii Camera3D bola použitá funkcia z aplikácie 3D Steroid, ktorá ako jediná z aplikácií v prieskume, zobrazuje pomocou akcelerometra horizontálnu rotáciu zariadenia 24.

Pre lepšie pochopenie senzorického navádzania je nutné najskôr popísať, v ktorých smeroch je zariadenie možné nakloniť. Zariadenie je možné nakláňať vo smeroch zobrazených na obrázku 6.



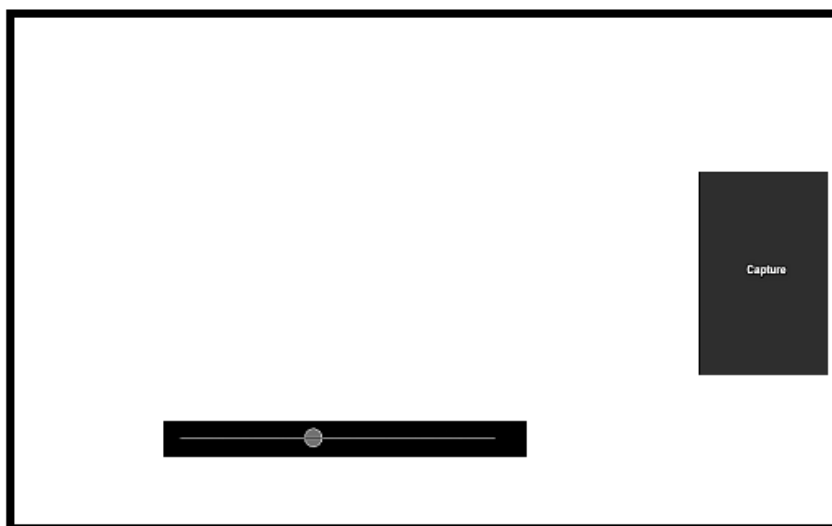
Obr. 6: Rotácie zariadenia na osách X,Y,Z

Ideálny okamih pre fotografovanie je v aplikácii 3D Steroid signalizovaný zmenou farby pohyblivej čiary na zelenú. Len v tej chvíli by používateľ mal vyfotografovať prvú aj druhú snímku stereoskopického páru.

Toto riešenie je veľmi elegantné, avšak stará sa len o správne rotácie zariadenia v jednom smere, a to v smere X. Čo ak ale používateľ naklonil zariadenie v smere Y alebo Z? Žiadna z

aplikácií z uvádzaného prieskumu sa o tieto pohyby nezaujíma, avšak aplikácia Camera3D rieši problém rozdielného a nechceného naklonenia počas snímania vo všetkých smeroch.

**6.2.2.1 Navádzanie pri snímaní prvej fotografie** Pri snímaní prvej fotografie aplikácia Camera3D pomocou objektu, ďalej v práci nazývaného guľička, používateľovi indikuje pohyb zariadenia po ose X na základe údajov z Akcelerometra zabudovaného v zariadení. 7.

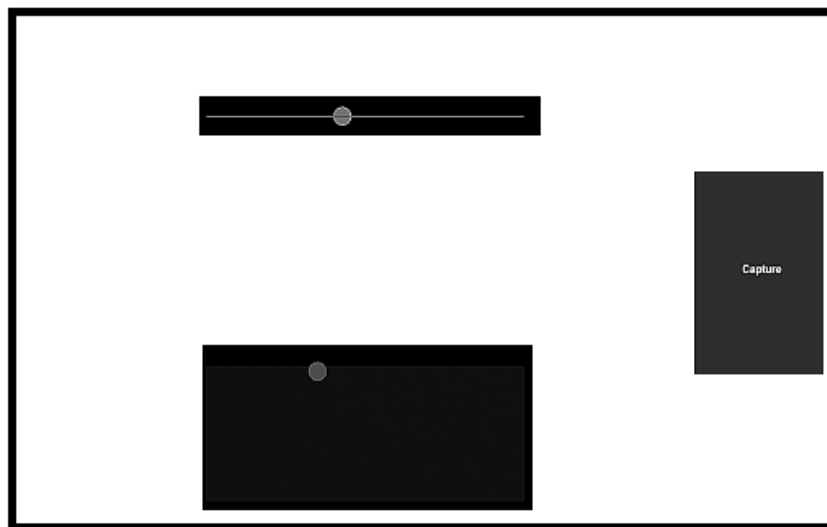


Obr. 7: Zobrazenie pohybu zariadenia na ose X

V prípade ak aplikácia od akcelerometrického senzora prijme údaj, že zariadenie je naklonené na ose X v ľubovoľnom smere, guľička zobrazená na spodnej strane displeja sa vydá do pohybu smerom, ktorým je toto zariadenie naklonené a zároveň bliká červenou farbou. Vo chvíli, keď používateľ upraví naklonenie zariadenia na ose X do polohy paralelnej so zemou, guľička sa zastaví a svieti na zeleno. Týmto spôsobom je používateľ zreteľne informovaný o presnom naklonení jeho zariadenia a tým sa vyvaruje nechcenej vzájomnej rotácii snímok, a zároveň neprírodzenému zobrazeniu reality, kedy napríklad voda tečie do kopca atp. Pri snímaní prvej snímky nie je dôvod sledovať naklonenie zariadenia na osách Y a Z nakoľko sú tieto smery ponechané na ľubovôli používateľa.

**6.2.2.2 Navádzanie pri snímaní druhej fotografie** Po vytvorení prvej snímky nastáva chvíľa, keď sa do procesu vytvorenia stereoskopického páru zapája aj monitorovanie zariadenia na osách Y a Z. V momente vytvorenia prvej snímky si aplikácia vyžiada od senzorov presnú polohu zariadenia vo všetkých smeroch, tzn. polohu na všetkých troch osách a uloží si ju. Následne sa zobrazí zložitejšie okno 8

Na spodnej strane displeja je guľičke umožnený pohyb v ďalšom rozmere, a tým možnosť pohybovať sa po ploche obdĺžnika. Tento pohyb reprezentuje naklonenie zariadenia na osách X a Y, pričom cieľom používateľa je zastavenie pohybu guľičky, a tým získanie záruky, že sa



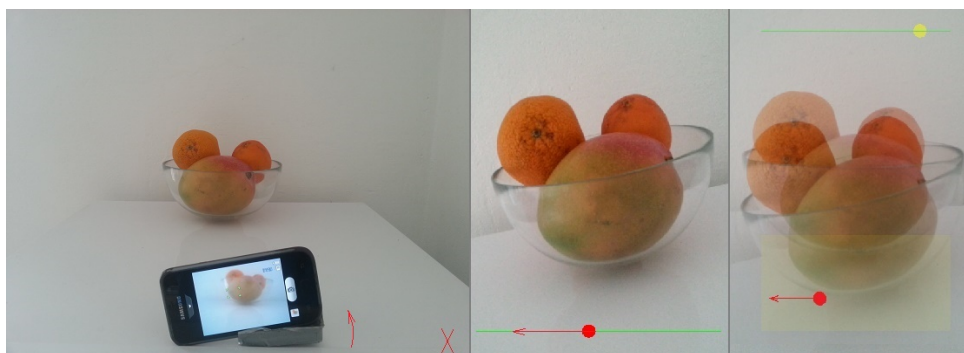
Obr. 8: Zobrazenie pohybu zariadenia na osách X, Y, Z

zariadenie nachádza v rovnakej X, Y pozícií ako bolo v momente zosnímania prvej fotografie. Zároveň v aplikácii prebieha ďalší proces. Na hornej strane displeja sa po nasnímaní prvej snímky zobrazí ďalšia guľička, ktorá reprezentujúca pohyb na ose Z pomocou magnetometra, ktorý funguje podobne ako kompas. Pri snímaní druhej snímky guľička zostáva nehybná v prípade, ak pozícia zariadenia na ose Z zodpovedá pozícií zariadenia v okamihu zosnímania prvej snímky. Používateľ aplikácie je veľmi dobre informovaný o pozícií svojho zariadenia pri snímaní prvej aj druhej snímky, je už len na jeho šikovnosti, či dokáže v rukách udržať zariadenie v presnej pozícií a vytvorí stereoskopický pár k vlastnej spokojnosti. Senzorické navádzanie síce zabezpečuje v plnej miere správne naklonenie zariadenia vo všetkých smeroch, avšak to ešte nie je záruka, že používateľ vytvorí ideálny stereoskopický pár. A to z jednoduchého dôvodu. Nie je náročné predstaviť si situáciu, kedy používateľ vytvoril prvú snímku stereoskopického páru na ktorej sa nachádzal napríklad jeho obľúbený koláč v kaviarni, neskôr kráčal k svojmu autu, pričom zariadenie stále držal v ideálnej X,Y, a Z pozícií a na druhej snímke vyfotografoval svoje auto. Na základe tohoto prehnane absurdného príkladu je zrejmé, že správna pozícia zariadenia ešte neznamená zaručený úspech. Aby používateľ zachytil na oboch snímkach rovnaký predmet a to aj z rovnakej polohy a výšky zariadenia, posunutý len a len o vzdialenosť zodpovedajúcu deviácií snímky, aplikácia ponúka ďalšiu funkciu 6.2.4.

**6.2.2.3 Nevhodný posun fotoaparátu** Nasleduje popis nevhodného naklonenia, natočenia alebo rotácie pri snímaní druhej fotografie. V tejto podkapitole je vysvetlenie akým spôsobom sú nevhodné rotácie zariadenia reprezentované pohybom guľičiek po displeji zariadenia počas snímania.

Nevhodné naklonenie na ose x 9 je reprezentované horizontálnym pohybom guľičky. Vo chvíli, kedy je zariadenie v správnej pozícií, tzn. pri snímaní prvej fotografie je to pozícia paralelná so

zemou, a pri snímaní druhej fotografie je to pozícia rovnajúca sa pozícií zariadenia pri snímaní prvej fotografie, tak je guľička statická a bez pohybu.



Obr. 9: Nevhodné naklonenie na ose X

Po nasnímaní prvej fotografie je pozícia na ose Y aplikáciou zapamätaná a pri snímaní druhej fotografie je nevhodné naklonenie smerom dopredu dole, tak ako na obrázku 10 reprezentované vertikálnym pohybom guľičky po displeji zariadenia. Úlohou používateľa je opäť zastavenie pohybu guľičky.



Obr. 10: Nevhodné naklonenie na ose Y

Po nasnímaní prvej fotografie je pozícia taktiež na ose Z aplikáciou zapamätaná a pri snímaní druhej fotografie je nevhodné naklonenie smerom vľavo a vpravo, tak ako na obrázku 10 reprezentované horizontálnym pohybom guľičky na vrchnej strane displeja zariadenia. Úlohou používateľa je opäť zastavenie pohybu guľičky.



Obr. 11: Nevhodné naklonenie na ose Z

### 6.2.3 Vytvorenie 3D snímky z vyfotografovaného stereoskopického páru

Hlavnou ideou celej aplikácie Camera3D je vytvorenie 3D snímky v niektorej zo známych a používaných vizualizačných techník. V ľubovoľnej chvíli po vytvorení stereoskopického páru (pred úpravou, po úprave), má používateľ možnosť vyrobiť 3D snímku vo vizualizačnej technike nazývanej Anaglyf alebo v technike nazývanej Side-by-side. Tieto novo vytvorené snímky sú okamžite zobrazené na displeji zariadenia, kde ich môže používateľ skontrolovať a zväžiť ďalšie úpravy prípadne uloženie na kartu SD. Dalo by sa povedať, že hrubým vstupom do aplikácie Camera3D je nasnímanie stereoskopického páru a výstupom aplikácie je upravená 3D snímka vo formáte Anaglyf alebo Side-By-Side.

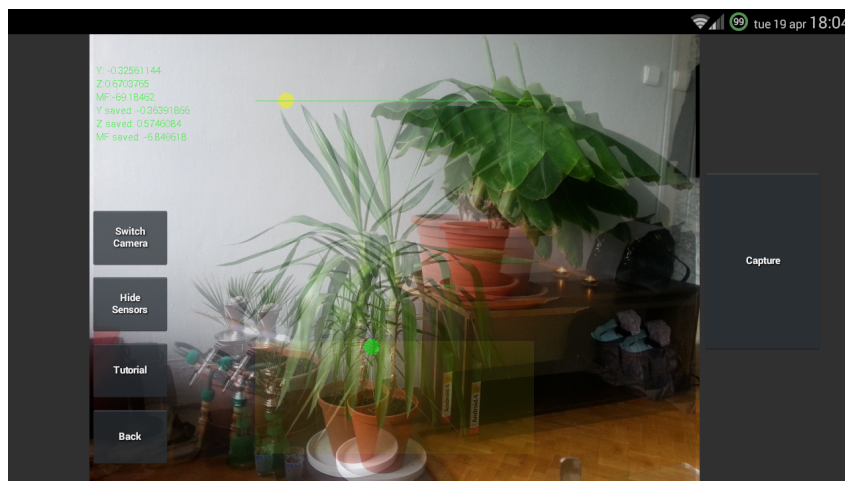
**6.2.3.1 Použité vizualizačné techniky** Ako bolo uvedené v kapitole 3.6.2, anaglyf je vizualizačná technika obľúbená vďaka jednoduchosti pozorovania, keď používateľovi stačí nasadiť lacné papierové 3D okuliare a okamžite si môže užiť svoje vlastné snímky v priestorovom zobrazení 12. Použitie tejto funkcie v aplikácii Camera3D bolo veľmi žiadúce a jej neprítomnosť by v používateľoch mohla vyvolať zmätočné reakcie, pretože pre väčšinu ľudí, je táto 3D vizualizačná technika jediná, ktorú poznajú. Druhou vizualizačnou technikou použitou v aplikácii Camera3D je Side-by-side 12, ktorá je charakteristická, ako je popísané v 3.6.2 tým, že okrem možnosti jej pozorovania v stereoskope, modernejší stereoskop je Google Cardboard, je možné užiť si priestorového zážitku aj bez akýchkoľvek pomôcok. Všetko čo je potrebné je len trošku skúsenejšie oko pozorovateľa, ktorý ovláda pohyb svojich očí a ich jednoduché pre-ostrenie a nasmerovanie. Viac k implementácii týchto funkcií bude popísané v kapitole 8.



Obr. 12: Vľavo anaglyf, vpravo Side-By-Side

### 6.2.4 Transparentné zobrazenie prvej snímky počas snímania druhej

Na obrázku 13 je snímka obrazovky zariadenia popisujúca okamih po vytvorení prvej snímky. Pre názorné vysvetlenie funkcie je možné povedať, že na obrazovke akoby zostal polo-priehľadný duch prvej snímky vďaka ktorému je používateľ jasne zoznámený s tým, čo má zachytiť na snímke druhej v prospech dokonalej 3D snímky.



Obr. 13: Zobrazenie prvej polo priehľadnej snímky

### 6.2.5 Korekcia stereoskopického páru

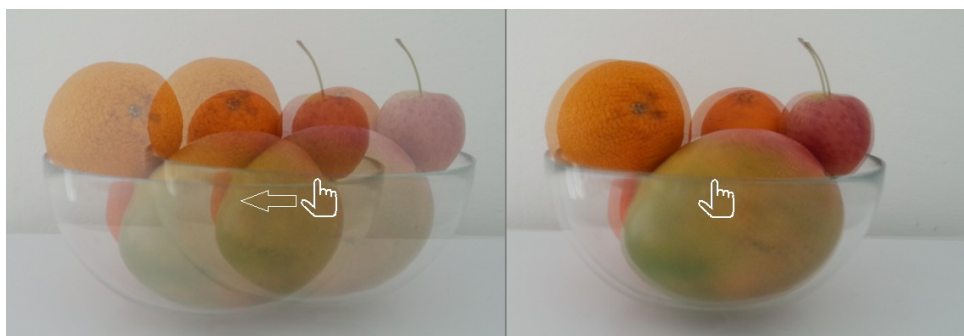
Ak by napriek všetkým doplnkovým funkciám, ktoré používateľovi pomáhajú pri snímaní, používateľ vytvoril stereoskopický pár, ktorý by bol nejakým spôsobom vzájomne nevhodne naklonený, znamenalo by to, že nerešpektoval vizualizáciu pri snímaní. Je pravdepodobné, že ak existuje vzájomne naklonenie stereoskopického páru, tak z výslednej 3D fotografie pozorovanej pomocou akejkoľvek z 3D vizualizačných techník, nebude mať pozorovateľ dobrý priestorový zážitok. Mohli by ho bolieť oči, alebo hlava, ba dokonca by nemuselo vôbec dôjsť k spojeniu fotografií a priestorovému zážitku. Aplikácia Camera3D ponúka niekoľko možností pre vylepšenie vytvoreného stereoskopického páru. V momente kedy používateľ dokončil fotografovanie oboch snímok, na displeji sa zobrazí náhľad na snímku vo vizualizačnej technike nazývanej Side-By-Side. Skúsenejší fotograf si dokáže spojiť stereoskopický pár v tomto formáte aj bez použitia stereoskopu a skontrolovať tak či sa mu podarilo vyfotografovať kvalitný stereoskopický pár. Menej skúsenejší fotograf vloží zariadenie do stereoskopu, alebo použije 3D red-cyan okuliare na skontrolovanie nového anaglyfu. Ak sa používateľovi zdá, že snímky nie sú dokonale spojené, aplikácia ponúka nasledujúce možnosti vylepšenia stereoskopického páru.

**6.2.5.1 Korekcia naklonenia** Túto funkciu je vhodné použiť v prípade, ak je zistené, že sa na niektorej zo snímok stereoskopického páru nachádzajú objekty v rozdielnej výške od spodného okraja snímky ako na druhej snímke. Takto vytvorené snímky je do určitej miery možné upraviť do správnej podoby najmä ak objekty na snímke boli snímané z prednej strany a nie príliš z nadhľadu. Na obrázku 27 je ukázané ako dokáže táto funkcia aplikácie Camera3D spolu s polo-automatickým spojením skvalitniť nevhodne vytvorený stereo-pár.

V tomto režime používateľ vidí ľavú snímku na pozadí zariadenia pričom je prekrývaná pravou polo-priehľadnou snímkou, čo je veľmi používateľsky prívetivé. Po priložení prsta na



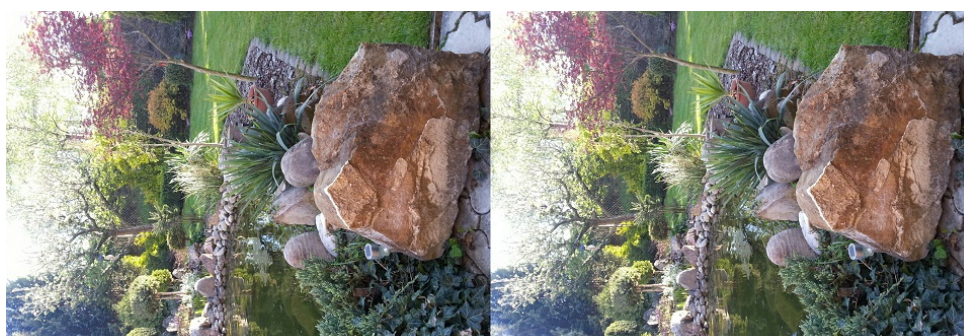
displej zariadenia má používateľ možnosť tieto snímky vzájomne vycentrovať, ako je ukázané na obrázku 14



Obr. 14: Manuálna korekcia pozície snímok

Veľkým prínosom tejto funkcie je, že okrem korekcie naklonenia snímok súčasne umožňuje manuálne spojenie snímok stereoskopického páru.

**6.2.5.2 Korekcia rotácie** Aplikácia Camera3D je navrhnutá na snímánie fotografií na šírku obrazovky zariadenia, to však neznamená, že neumožňuje vytvorenie 3D snímky zo snímok, ktoré boli nasnímané na výšku. Aplikácia Camera3D ponúka možnosť korekcie nevhodnej rotácie na základe užívateľského vstupu tak, ako je zobrazené na nasledujúcich snímkach 15,16.



Obr. 15: Nevhodne vyrotovaný stereoskopický pár, vytvorený snímaním na výšku zariadenia

#### 6.2.6 Polo-automatické spojenie stereoskopických snímok

Metóda pre polo-automatické spojenie snímok je pokročilá funkcia aplikácie Camera3D 26. Používateľ je vyzvaný, aby určil v oboch snímkach bod zhody, ktorý sa vo výslednom stereoskopickom páre bude vzájomne prekrývať. Ideálne je aby sa tieto body nachádzali v približných stredoch snímok a zároveň aby sa jednalo o bod umiestnený na objekte, ktorý je jeden z najbližších k fotografovi, a preto je v aplikácii Camera3D približne do stredu snímky bod vopred predvolene nastavený. Na používateľovi už zostáva len presné zacielenie bodov. Táto metóda úpravy snímok je veľmi efektívna vzhľadom na to, že sa predpokladá, že vďaka senzorickému navádzaniu





Obr. 16: Vytvorený anaglyf s korekciou rotácie

používateľ vytvoril takmer dokonalý stereoskopický pár. Mohlo by sa zdať, že ideálnym by bolo použitie niektorej z metód automatického spojenia stereoskopického páru s detekciou zhodných bodov ba dokonca vykonanie rektifikácie snímok, avšak tým by robustnosť aplikácie Camera3D veľmi narástla a v prípade použitia OpenCV by bol používateľ nútený k inštalácii ďalšej aplikácie a to OpenCV manažéra čo by spolu s pravdepodobne pomalšími reakciami aplikácie nebolo vôbec žiadúce. Výpočtový čas automatického spojenia snímok s detekciou vzájomných spoločných bodov sa aj na desktopových aplikáciach pohybuje v sekundách, a preto nepripadalo v úvahu použitie niektorej z týchto techník popísaných v kapitole na mobilnom zariadení s obmedzenou operačnou pamäťou a nie príliš výkonným procesorom. Implementovaná polo-automatická metóda v aplikácii Camera3D je preto ideálnym kompromisom medzi rýchlosťou aplikácie a dokonalým spojením stereoskopického páru. Na nasledujúcej snímke je skutočná funkcia demonštrovaná. Ďalšou veľkou výhodou metódy polo-automatického spojenia je, že používateľ je nútený pri vyberaní spoločných bodov skúmať v snímkach hlboké detaily, a preto často odhalí prípadnú neostrosť v niektorej zo snímok a uvedomí si, že v prospech vytvorenia kvalitného stereoskopického páru bude musieť nasnímať nový a kvalitnejší stereo pár.

#### 6.2.7 Exportovanie vytvorenej snímky na kartu SD

Aby používateľ skutočne vyrobil vlastnú 3D snímku a mohol s ňou ďalej ľubovoľne nakladať, je funkcia uloženia na SD kartu nutnou podmienkou. Výhodou aplikácie Camera3D oproti ostatným aplikáciám z prieskumu je, že ukladá snímky vo výrazne vyššej kvalite. Vyššie rozlíšenie snímky priaznivo vplýva na samotný 3D efekt, takže táto funkcia je jedným zo zásadných prínosov aplikácie Camera3D.

### **6.2.8 Zobrazenie uložených snímok v galérii**

Po uložení snímok na kartu SD má používateľ možnosť pomocou tlačidla vyvolať jeho obľúbenú aplikáciu na prezeranie mediálnej galérie.

### **6.2.9 Zoznámenie používateľa s aplikáciou - tutoriál**

Po prvom spustení aplikácie je používateľovi doporučované pozorné preštudovanie obrazového tutoriálu, v ktorom sa zoznami nielen so základmi vytvárania stereoskopických párov, ale aj s tým ako funguje senzorické navádzanie a urobí z na prvý pohľad komplikovaných a nekontrolovateľne sa pohybujúcich objektov na obrazovke, svojich pomocníkov na ceste za vytúženou 3D fotografiou.

## **6.3 Nefunkčné požiadavky**

Pri návrhu aplikácie bolo jedným zo základných cieľov, aby aplikácia reagovala bez oneskorenia, a aby používateľ nemusel inštalovať žiadne doplnkové aplikácie k jej používaniu. Ďalšími kritériami aplikácie bola grafická prívetivosť a zdanlivá jednoduchosť zakrývajúca zložitejšie výpočty prebiehajúce v aplikácii.

- Jednoduchosť ovládania
- Grafická prívetivosť
- Funkčnosť bez nutnosti inštalácie ďalších aplikácií
- Okamžitá pohotovosť
- Optimalizácia pre fungovanie na viacerých zariadeniach

## 7 Analýza

Aplikácia Camera3D je vytvorená pre platformu Android, pričom výpočtové jadro aplikácie, v ktorom sú pokročilé metódy vytvárania 3D snímky napísane v C a Java. Knižnica je v samostatnom balíčku, vďaka čomu je predpripravená na použitie v ďalšej ľubovoľnej platforme. Cieľovou skupinou aplikácie sú nie len skúsení fotografi, ktorí ocenia presnosť jej senzorového navádzania a možnosti upravenia spojenia stereoskopického páru, ale aj začiatočníci, ktorí sa s procesom výroby 3D fotografie ešte nestretli.

## 8 Implementácia výpočtovej knižnice

Jedným z cieľov tejto práce bolo vytvorenie knižnice, ktorej obsahom je výpočtové jadro aplikácie. Knižnice sú implementované z dôvodu ľahkého znovupoužitia v iných aplikáciách. Dalo by sa povedať, že do knižnice boli zahrnuté všetky algoritmy, ktorých potenciálna doba výpočtu bola väčšia ako okamžitá. Knižnicu je možné rozdeliť na metódy napísané v jazyku C a v jazyku Java.

### 8.1 Metódy knižnice napísané v jazyku C

Volanie natívnych metód písaných v jazyku C je ideálnym riešením pre Aplikáciu Camera3D. Vo chvíli kedy je potreba prechádzať dve veľké bitmapy pixel po pixely. Podľa štúdie [22] je použitie NDK jedinečnou príležitosťou pre zrýchlenie výpočtov v aplikáciách. Vďaka použitiu JNI sa veľmi výrazne zvýši výpočtový výkon funkcií výpočtu vizualizačných techník anaglyfu a side-by-side nad dvoma bitmapami.

#### 8.1.1 Výpočet bitmapy anaglyfu

Funkcia výpočtu anaglyfu zo stereoskopického páru je v prvom rade deklarovaná v jave ako natívna metóda pomocou nasledujúceho kódu:

---

```
private static native void calculateAnaglyph(Bitmap imgTrueAnaglyph, Bitmap
    imgLeft, Bitmap imgRight);
```

---

Výpis 2: Deklarovanie natívnej metódy calculateAnaglyph()

Vstupom metódy sú tri bitmapy. Bitmapy imgLeft a imgRight predstavujú snímky stereoskopického páru a bitmapa imgTrueAnaglyph je prázdna bitmapa s rovnakou šírkou a výškou ako majú bitmapy stereoskopického páru. Pre efektivitu metódy je vhodné aby jej návratový typ bol void. Výsledná bitmapa anaglyfu teda nie je metódou navrátená, ale len upravená. Pokiaľ by metóda calculateAnaglyph bola volaná v tejto situácii bez dodania natívneho kódu tak by volanie zlyhalo s výnimkou Java.lang.UnsatisfiedLinkError. Ďalším krokom je vygenerovanie hlavičkového súboru s deklaráciami pre jazyk C pomocou konzolového nástroja Javah, ktorý je súčasťou JDK. Javah pracuje nad skompilovanými triedami, treba mu len predať classpath s umiestnením týchto tried. Nasleduje popis natívneho kódu metódy calculateAnaglyph v jazyku C.

---

```
uint32_t trueAnaglyph(uint32_t imgLeft, uint32_t imgRight){
    uint32_t x;
    uint32_t pom3 = (imgRight)& 0xFF;
    uint32_t pom2 = (imgRight) & 0xFF;
    uint32_t pom1 = (imgLeft) & 0xFF;
    x = pom1 << 0 | pom2 << 8 | pom3 << 16;
```

```

    return x;
}

```

---

Výpis 3: Kód natívnej metódy calculateAnaglyph()

Vďaka tejto metóde je vypočítaný každý jedinečný pixel nového výsledného anaglyfu, ktorý obsahuje červenú farebnú zložku z ľavej bitmapy stereoskopického páru, a súčasne modrú a zelenú farebnú zložku pravej bitmapy stereoskopického páru.

Bol použitý nasledovný vzorec pre výpočet farebného anaglyfu: 1 použitý z [26]

$$\begin{pmatrix} r_a \\ g_a \\ b_a \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix} \quad (1)$$

Pomocou vnorených cyklov sú prejdene všetky pixely pôvodnej ľavej a pravej bitmapy a tieto sú jeden po druhom zapísané do predom pripravenej bitmapy so zhodujúcou sa šírkou aj výškou pôvodnej bitmapy. Na záver je nutné uvoľnenie pixelov z pamäte pomocou príkazu `unlockPixels()`.

---

```

AndroidBitmap_unlockPixels(env, imgLeft);
AndroidBitmap_unlockPixels(env, imgRight);
AndroidBitmap_unlockPixels(env, imgTrueAnaglyph);

```

---

Výpis 4: Uvoľnenie pixelov z pamäte

### 8.1.2 Výpočet 3D snímky vo formáte Side-By-Side

Vyrobenie výslednej snímky pre zobrazenie vo vizualizačnej technike Side-By-Side je jednoduchšie v zmysle logiky algoritmu v porovnaní s vyrobením Anaglyfu, avšak časovo náročnejšie, pretože šírka výslednej bitmapy má dvojnásobnú veľkosť. Algoritmus napísaný v jazyku C vloží dve snímky stereoskopického páru pixel po pixely vedľa seba do pripravenej bitmapy o rovnakej výške avšak dvojnásobnej šírke v porovnaní s pôvodnou bitmapou.

## 8.2 Metódy napísané v jazyku Java okrajovo používajúce knižnicu

Obsahom knižnice sú aj dve funkcie napísané v jazyku Java. Výpočtová zložitosť týchto metód nie je tak vysoká, aby v tomto prípade použitie NDK zrýchlilo daný výpočet.

### 8.2.1 Manuálna editácia stereoskopického páru

Princípom metódy manuálnej editácie stereoskopického páru je, že sa porovnávajú vzájomné pozície jednotlivých CustomView a následne sa bitmapy orežú z príslušných strán, tak aby bolo zachované používateľom nastavené prekrytie. Premenné `diffLeft` a `diffTop` inicializované v nasledujúcej ukážke zdrojového kódu udržiujú údaj o počte pixelov, o ktoré sú CustomView

vzájomne posunuté v horizontálnom a vertikálnom smere a práve tieto hodnoty sú predané metóde z výpočtovej knižnice, ktorá vráti žiadané výsledky.

---

```
int diffLeft = sv2.getLeftPos() - sv.getLeftPos();  
int diffTop = sv2.getTopPos() - sv.getTopPos();
```

---

Výpis 5: Vzájomný posun dvoch CustomView

Následne je dôležité zistiť šírku a výšku jednotlivých bitmáp

---

```
int width = bitmaps[0].getWidth();  
int height = bitmaps[0].getHeight();
```

---

Výpis 6: Zistenie šírky a výšky bitmáp

Logika prípravy bitmáp na spojenie je nasledovná. Používateľovi je umožnené posúvať jednu z bitmáp, tak aby vycentroval ich vzájomnú pozíciu. Ak bola pozícia pohyblivej bitmapy upravená o určitý počet pixelov smerom doprava, tak táto bitmapa bude zprava orezaná o rovnaký počet pixelov, a zároveň statická bitmapa bude orezaná o rovnaký počet pixelov zľava. Podobná logika je použitá aj pri posúvaní bitmapy v ostatných smeroch. Ukážka zdrojového kódu je k dispozícii v prílohe diplomovej práce. 29

### 8.2.2 Riešenie problému zobrazenia snímky na obrazovke zariadenia.

Keďže rozmery snímky sú výrazne väčšie ako rozmery obrazovky bežného mobilného telefónu, bolo nutné prispôbiť ich zobrazenie na obrazovke. Nebolo problémom nastaviť náhľady na fotografie tak, aby ich veľkosť zodpovedala rozmerom displeja, avšak problém nastal vo chvíli manuálnej korekcie týchto snímok, kedy má používateľ možnosť meniť ich vzájomnú pozíciu a tým pripraviť snímky na orezanie. Vzájomný posun snímok je monitorovaný v pixeloch a je zrejmé, že posun obrázkov, ktoré sú kvôli veľkosti displeja zmenšené nebude zodpovedať reálnej veľkosti snímok. Preto je nutné dodatočné dopočítanie popísané nasledovným zdrojovým kódom, ktorý je spoločný jak pre funkciu manuálnej korekcie, tak pre polo-automatické spojenie stereoskopického páru 7. V budúcnosti by bolo možné implementovanie funkcie približovacej lupy, v prípade, že by sa po publikácii aplikácie používatelia sťažovali na náročnosť presného určenia pozície bitmáp.

### 8.2.3 Polo-automatické spojenie stereoskopického páru

V aplikácii je obrazovka rozdelená na šírku, na polovicu. Na ľavej strane obrazovky sa nachádza ľavá snímka stereoskopického páru a na pravej naopak. Snímky sú vložené do Custom View s ktorým je možné pohybovať po kresliacom plátne Canvas. V strede každej snímky je pomocou metódy onDraw() vykreslený zelený bod a používateľ pomocou neho určí zhodné body v oboch snímkach, vďaka čomu polo-automatická metóda z volaná z knižnice oreže jednotlivé snímky podobne, ako tomu bolo v prípade manuálnej editácie a tým ich pripraví na spojenie. Je použitý

rovnaký zdrojový kód orezania jednotlivých snímok stereoskopického páru ako v predchádzajúcej kapitole. 29

Aby vykreslené snímky zobrazené v Custom View veľkosťou zodpovedali rozmerom obrazovky zariadenia je nutné vytvoriť zmenšené, náhľadové kópie, ktoré sú zobrazené používateľovi. Pri samotnom centrovaní snímok sa porovnávajú ich vzájomné pozície, avšak z dôvodu zmenšenia snímok je nutné dodatočné prepočítanie ich vzájomného posunu na veľkosť zodpovedajúcu veľkosti snímok pred zmenšením podobne ako v kapitole 8.2.1.

---

```
float aspectRatio = photoWidth / photoHeight;
int height1 = dispheight;
int width1 = Math.round(height1 * aspectRatio);
reductionRatioBitmap = photoHeight / (float)height1;
Bitmap scaledBitmap1 = Bitmap.createScaledBitmap(bitmaps[0], width1,
    height1, false);
Bitmap scaledBitmap2 = Bitmap.createScaledBitmap(bitmaps[1], width1,
    height1, false);
```

---

Výpis 7: Práca s rozmermy bitmáp

Následný prepočet do veľkosti skutočnej bitmapy je vykonaný pomocou vynásobenia premennej reductionRatioBitmap s posunom zmenšenej snímky po obrazovke zariadenia.

## 9 Implementácia aplikácie Camera3D

K samotnej implementácii aplikácie bolo pristupované iteratívnym spôsobom. Na počiatku vývoja boli namiesto samotného snímania snímok používané bitmapy priamo vložené do programu. Až neskôr bolo do aplikácie naprogramované snímanie pomocou fotoaparátu zabudovaného v zariadení a ďalšie funkcie.

Ako bolo uvedené skôr základom vývoja aplikácie Camera3D bolo napodobňovanie obľúbených funkcií z existujúcich aplikácií, pričom však princíp týchto funkcií zostával neodhalený, pretože nebolo umožnené nahliadnutie do zdrojových kódov týchto aplikácií.

### 9.1 Hlavná aktivita

Moderné android programovanie je založené na fragmentoch. Môže sa to zdať prekvapivé, ale celá aplikácia Camera3D používa len jednu hlavnú aktivitu, ktorá ma ako layout nastavený LinearLayout s priradeným id názvom fragment container. Tento názov výstižne vypovedá o použití LinearLayoutu ako kontajneru v ktorom sa pomocou nasledujúceho zdrojového kódu menia fragmenty pripnuté k aktivite.

---

```
transaction.replace(R.id.picutre_fragment_container, AnaglyphFragment.  
    newInstance(bitmaps)).commit();
```

---

Výpis 8: Práca v java s fragmentmi

Metóda `.replace` pochádza z triedy `FragmentTransaction` zo základného balíčka android API. Jej funkciou je výmena fragmentu pripnutého k aktivite. V podstate ako keby bola na starý fragment zavolaná metóda `.remove(Fragment)` a následne na nový fragment zavolaná metóda `.add(Fragment)`.

### 9.2 Použitie fragmentov

V prvej iterácii návrhu aplikácie nebolo plánované celoaplikačné používanie fragmentov, avšak ukázalo sa, že medzi aktivitami nie je možné predávať tak veľké dáta, ako sú dve čerstvo zosnímané snímky z kvalitných zadných fotoaparátov dnešných mobilných zariadení. Použitie fragmentov bolo teda ideálnou a nevyhnutnou voľbou. Ak by programátor chcel predať pole veľkých bitmáp medzi aktivitami, nezostávalo by mu nič iné, ako v jednej aktivite jednotlivé snímky uložiť na kartu SD a v druhej aktivite si tieto snímky vyzdvihnúť. Fragment musí byť pripojený k aktivite pomocou funkcie `.onAttach` nakoľko by mohlo dôjsť k nechcenému prerušeniu aktivity a tým strateniu prístupu k fragmentu. Vo chvíli keď už je fragment nepotrebný, je naň zavolaná metóda `.onDetach` pomocou ktorej sa od aktivity odpojí a tým je zároveň recyklovaný z pamäte.



## 9.3 Snímanie fotografií

Android ponúka snímanie fotografií v aplikácií na zariadení dvomi spôsobmi. Prvý, a zároveň jednoduchší je popísaný v kapitole 2.3, kedy vývojár využije možnosti snímania Android vstavanou aplikáciou. Druhý a zložitejší spôsob je vytvorenie vlastnej aplikácie pre fotografovanie čo vývojárovi dáva príležitosť urobiť veci inak, ako sú štandardne dané operačným systémom Android. Nasleduje podrobný popis krokov potrebných pre vytvorenie vlastnej aplikácie snímajúcej snímky fotoaparátom.

### 9.3.1 Overenie existencie fotoaparátu

Je potrebné vytvorenie metódy pre overenie existencie fotoaparátov a zažiadanie o prístup k nim.

---

```
private boolean hasCamera(Context context) {  
    //check if the device has camera  
    if (context.getPackageManager().hasSystemFeature(PackageManager.  
        FEATURE_CAMERA)) {  
        // this device has a camera  
        return true;  
    } else {  
        // no camera on this device  
        return false;  
    }  
}
```

---

Výpis 9: Overenie existencie fotoaparátov

Keďže snímanie fotografií, je nutnou funkciou mobilných zariadení, ktoré chcú aplikáciu Camera3D nainštalovať, je nutné pridať povolenie do Android Manifestu a to nasledovným spôsobom.

---

```
<manifest ... >  
    <uses-feature android:name="android.hardware.camera"  
        android:required="true" />  
    ...  
</manifest>
```

---

Výpis 10: Ukážka časti Android Manifestu

Tým je zaistené, že ak by bola aplikácia publikovaná na obchode Play store, tak pri žiadosti o stiahnutie aplikácie zo strany používateľa, ktorého zariadenie nemá fotoaparát, nebude aplikácii umožnená inštalácia na toto zariadenie. Z bezpečnostných dôvodov je ale vhodné súčasne aj použitie funkcie na overenie existencie fotoaparátu v samotnej aplikácií, pretože môže nastať

prípád, kedy samotný fotoaparát prestane reagovať, je takpovediac odpojený a tým by došlo k pádu aplikácie Camera3D pri jej spustení, čo by bolo nežiadúce.

### 9.3.2 Výber fotoaparátu

Ak bola úspešne zistená, dostupnosť fotoaparátu, tzn. funkcia `.hasCamera` vrátila boolean hodnotu `true`, ďalším krokom je žiadosť o prístup k fotoaparátu vytvorením inšancie triedy `Camera`. Pristupovanie k fotoaparátu je vykonané pomocou metódy `Camera.open()`. Nasleduje zdrojový kód výberu predného respektíve zadného fotoaparátu.

---

```
public void chooseCamera() {  
    //if the camera preview is the front  
    if (cameraFront) {  
        int cameraId = findBackFacingCamera();  
        if (cameraId >= 0) {  
            mCamera = Camera.open(cameraId);  
            mPicture = getPictureCallback();  
            mPreview.refreshCamera(mCamera);  
        }  
    } else {  
        int cameraId = findFrontFacingCamera();  
        if (cameraId >= 0) {  
            mCamera = Camera.open(cameraId);  
            mPreview.refreshCamera(mCamera);  
        }  
    }  
}
```

---

Výpis 11: Vytvorenie inšancie triedy `Camera`

### 9.3.3 Vytvorenie náhľadu

Pre používateľa je dôležité aby pri snímaní fotografie videl to, čo je v zábere fotoaparátu zariadenia. Náhľadová trieda, ktorá je schopná naživo zobrazíť dáta prichádzajúce z fotoaparátu je vytvorená nasledujúcim kódom.

---

```
public class CameraPreview extends SurfaceView implements SurfaceHolder  
    .Callback
```

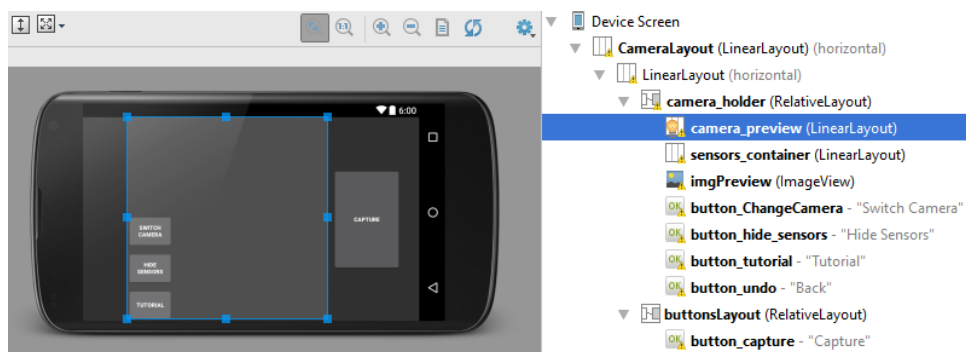
---

Výpis 12: Vytvorenie náhľadovej triedy

Je nutné, aby trieda dedila z rodičovskej triedy `SurfaceView` a implementovala rozhranie `SurfaceHolder.Callback` v zmysle zachytenia spätného volania udalostí pre vytváranie a ničenie náhľadu, ktoré sú potrebné pre priradenie vstupu fotografického náhľadu.

#### 9.3.4 Umiestnenie náhľadu do layoutu

Trieda `CameraPreview` ktorá bola skôr vytvorená, musí byť následne umiestnená do Android layoutu spolu s ostatnými používateľskými ovládacími prvkami potrebnými pre snímanie fotografií, ako sú tlačidlá, navádzacie senzory apod na obrázku 17



Obr. 17: Rozloženie layoutu pri snímaní

#### 9.3.5 Snímanie fotografií

Po tom, čo bol vytvorená náhľadová trieda a view layout v ktorom bude zobrazená je všetko pripravené na snímanie fotografií. V aplikácii `Camera3D` bol nastavený listener na tlačidlo snímania a po stlačení tohoto tlačidla sa zavolá metóda `Camera.takePicture()`, ktorá prijíma tri parametre `public final void takePicture (Camera.ShutterCallback shutter, Camera.PictureCallback raw, Camera.PictureCallback jpeg)` a vráti data z fotoaparátu.

---

```
View.OnClickListener captureListener = new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        mCamera.takePicture(null, null, mPicture);  
    }  
};
```

---

Výpis 13: Snímanie fotografie

Po nasnímaní obrázku, ako posledný krok vo chvíli, kedy už Aplikácia `Camera3D` fotoaparát nepoužíva je nutné uvoľniť kameru pomocou metódy `Camera.release()`.

## 9.4 Indikácia vhodného okamžiku snímania - zobrazenie senzorickej navigácie

Obrázok 17 popisuje rozloženie layoutu použitého pri snímaní fotografie. Lineárny layout s priradeným id sensorsContainer sa vzájomne prekrýva spolu s ostatnými prvkami vloženými do relatívneho layoutu s priradeným id cameraHolder. Do sensorsContaineru je pridaná inštancia vlastnej, užívateľsky vytvorenej triedy SensorsView1. Nasleduje ukážka zdrojového kódu.

---

```
((LinearLayout) getActivity(). findViewById(R.id.sensors_container)).addView(sensorsView1);
```

---

Výpis 14: Pridanie sensorsView1 do sensorsContaineru

## 9.5 Ovládanie pohybu guľičiek na základe senzorov

SensorsView je trieda, ktorá má ako svojho predka triedu View a prepisuje metódu onDraw vďaka čomu je možné kresliť 2D objekty priamo na Canvas. Guľičky reprezentujúce pohyb zariadenia vďaka informáciám od senzorov sú vykresľované na dvojrozmerný Canvas a ten prekrýva aktuálny náhľad z fotoaparátu. V triede SensorsView1 je implementovaná metóda onSensor, ktorá nastavuje hodnoty triednych premenných na základe pozície zariadenia, pričom je volaná vždy, keď sa zmení niektorá z hodnôt. Nasleduje implementácia metódy onSensor() z triedy SensorsView.

---

```
public void onSensor(SensorEvent event){  
    if (event.sensor == accelerometerSensor) {  
        x = event.values[0];  
        y = event.values[1];  
    }else if (event.sensor == magneticFieldSensor) {  
        z = event.values[0];  
    }  
}
```

---

Výpis 15: Implementácia metódy onSensor()

Guľička na Canvase je vykresľovaná vždy nanovo pomocou metódy drawCircle(), v momente, keď je hodnota triednej premennej (position) zmenená.

---

```
int height = canvas.getHeight();  
float radius = 12;  
position = position+(y*speed);  
canvas.drawCircle(position, height - 100, 12, paint);
```

---

Výpis 16: Pohyb guľičky

Konštanta height reprezentuje výšku Canvasu, ktorý je rozťahnutý na fullscreen. Konštanta radius predstavuje priemer vykresľovanej guľičky. Premenná position reprezentuje pozíciu, ktorá

sa mení horizontálne na základe premennej `y` ktorá je získavaná od akcelerometrického senzora a je vynásobená konštantou `speed`, pre zrýchlenie pohybu guličky. Na základe tohoto princípu pôsobí vykresľovanie guličky na obrazovke zariadenia ako animovaný a veľmi plynulý pohyb, čo je pre používateľa vždy príjemné.

Po zhotovení prvej snímky je na obrazovke zariadenia nahradený `SensorsView1` za `SensorsView2` pomocou nasledujúceho zdrojového kódu.

---

```
((LinearLayout) getActivity(). findViewById(R.id.sensors_container)).  
    removeView(sensorsView1);  
((LinearLayout) getActivity(). findViewById(R.id.sensors_container)).addView(  
    sensorsView2);
```

---

Výpis 17: Senzory pri snímaní druhej fotografie

`SensorsView2` je zložitejší ako `SensorsView1` avšak pracuje na rovnakom princípe kedy na základe všetkých premenných (`x,y,z`) sú prekresľované pozície dvoch guľičiek umiestnených na obrazovke ako na obrázku.

## 9.6 Riešenie problémov spôsobených nízkou kapacitou operačnej pamäte mobilných zariadení

Pri implementácii aplikácie `Camera3D` vznikol problém pri používaní plného rozlíšenia fotografií, až 8mpx, vytvorených na kvalitnom zadnom fotoaparáte zariadenia Samsung galaxy S3. Rozmery vytvorených fotografií sú 3264x2448 a vo chvíli kedy aplikácia `Camera3D` vytvorila 3D snímku vo vizualizačnej technike `Side-By-Side` boli rozmery 3D fotografie s dvojnásobnou šírkou 6528x2448 čo je až 16 megapixelov. Štandardne pridelená RAM pre aplikácie sa líši od zariadenia k zariadeniu pričom sa pohybuje okolo 16 alebo 24MB, avšak Android umožňuje aplikáciám pridelenie väčšieho objemu operačnej pamäte, ak si o to aplikácia zažiada v Android manifeste pomocou nasledovného zdrojového kódu

---

```
android:largeHeap="true"
```

---

Výpis 18: Pridelenie väčšieho objemu operačnej pamäte v manifeste

V tomto prípade má aplikácia povolené čerpať výrazne väčší objem pamäte. Na testovanom zariadení Samsung Galaxy S3 hodnoty bezpečne prekračovali 150MB, avšak pri prekročení 200MB aplikácia spadla s chybou `OutOfMemoryError`, čo znamená prekročenie limitu RAM.

Kľúčovým bolo nájdenie kompromisu medzi spoľahlivosťou aplikácie a kvalitou snímaných snímok.

Štandardné a jednoduché Android riešenie pre zníženie rozlíšenia snímaných fotografií je použitie options a nastavenie `inSampleSize` na ľubovoľnú hodnotu mocnín čísla 2. Každý stupeň mocniny delí šírku a výšku pôvodnej snímky na polovicu, z čoho vyplýva, že delí počet megapixelov štyrmi. Takže pri nastavení `inSampleSize` na hodnotu 2 je snímka s 8megapixelmi zmenená

na snímku s 2 megapixelmi, čo je veľmi razantné zníženie kvality. Tento proces je zobrazený nasledujúcim zdrojovým kódom.

---

```
BitmapFactory.Options options = new BitmapFactory.Options();
options.inSampleSize = 2;
Bitmap bm = BitmapFactory.decodeByteArray(data, 0, data.length,
options);
```

---

Výpis 19: Štandardné zníženie kvality snímok

Zníženie kvality z 8 na 2 megapixely je veľmi výrazné a keďže Android neponúka iné štandardné riešenie, ktoré by nezmenšovalo snímky tak výrazne, bolo prístupné k vytvoreniu vlastného algoritmu na čiastočné zníženie kvality snímok:

---

```
Bitmap bm = BitmapFactory.decodeByteArray(data, 0, data.length, options);
float pictureQuality = 1.3F;
float width = bm.getWidth();
float height = bm.getHeight();
float ratio = width/(width/pictureQuality);
float newWidth = width / ratio;
float newHeight = height / ratio;
Bitmap bitmap = Bitmap.createScaledBitmap(bm, (int)newWidth , (int)
newHeight, false);
```

---

Výpis 20: Vlastný algoritmus pre zníženie rozlíšenia snímok

Algoritmus zabráňuje nechcenému prekročeniu prideleného objemu RAM a zároveň poňcháva snímky v čo najvyššej kvalite so zachovaním pomeru strán.

## 9.7 Uloženie 3D snímky na kartu SD

Procesy, ktorých dĺžka vykonania predstavuje neurčitý časový úsek, tzn. napr. viac ako desatinu sekundy a existuje určitá pravdepodobnosť, že tieto procesy budú počas vykonávania prerušené, nemali by byť spúšťané v hlavnom aplikačnom vlákne. AK by proces uloženia fotografie bežal v hlavnom vlákne aplikácie, priemerne tento proces trvá od 0 do 2 sekúnd, mohlo by sa stať, že počas tohto intervalu bude hlavné aplikačné vlákno pozastavené a tým proces uloženia nenávratne stratený, pričom používateľ by nemal prehľad o tom, či bol tento proces naozaj dokončený, alebo predčasne ukončený. Pozastavenie hlavného aplikačného vlákna nie je nič neobvyklé a môže nastať pri viacerých štandardných situáciach, ako je otočenie obrazovky zariadenia alebo príchod hovoru. Riešením je vloženie procesu uloženia fotografie do samostatného fragmentu a jeho pripojenie k aktivite pomocou funkcie `onAttach()`. Tým je zabezpečené, že po znovu obnovení hlavného aplikačného vlákna bude obnovený aj predtým prerušený fragment, a ak sú mu v prepísanej metóde `onResume()` správne doručené všetky inštancie, ktoré používal pred prerušením,

dokončenie procesu bude zaručene úspechom. Pre proces uloženia snímky v samostatnom vlákne bol použitý `AsyncTask()`, a v hlavnej metóde `doInBackground()` boli vykonané následovné kroky pre uloženie snímky na kartu SD.

- Vytvorenie novej zložky na karte SD- Je vhodné aby fotografie z aplikácie Camera3D boli ukladané do samostatnej zložky(súboru) aby používateľ mal možnosť tieto fotografie držať pohromade. Ak zložka s konkrétnym názvom ešte nebola na karte SD vytvorená, tak bude použitý nasledovný zdrojový kód pre jej vytvorenie.

---

```
File mediaStorageDir = new File("/sdcard/", "Camera3D");
```

---

Výpis 21: Vytvorenie zložky

- Vytvorenie časového razítka- Je nutné, aby bol názov každej fotografie unikátny voči ostatným, a preto použitie časového razítka, ktoré reprezentuje na sekundu presný časový okamžik, kedy bola fotografia vytvorená je zároveň zárukou, že žiadne dve fotografie nebudú rovnakého názvu a zároveň je z názvu fotografie možné prečítať užitočnú informáciu o čase jej vytvorenia.

---

```
String timeStamp = new SimpleDateFormat("yyyyMMdd_HH:mm:ss").format(  
    new Date());
```

---

Výpis 22: Vytvorenie časového razítka

- Nakoniec je možné pristúpiť k samotnému vytvoreniu mediálneho súboru a k jeho zápisu do úložiska. Pre demonštráciu nasleduje zjednodušený zdrojový kód bez ošetrovania výnimiek.

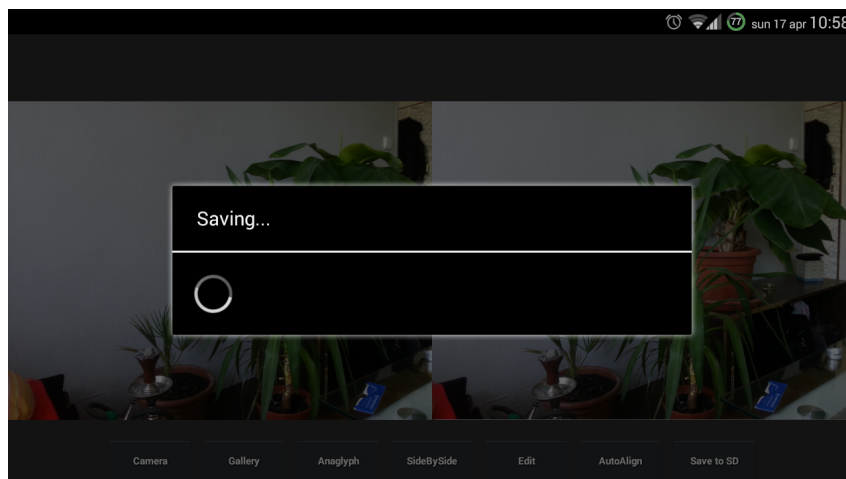
---

```
ByteArrayOutputStream stream = new ByteArrayOutputStream();  
imgTrueAnaglyph.compress(Bitmap.CompressFormat.JPEG, 100, stream);  
byte[] byteArray = stream.toByteArray();  
File pictureFile = new File(mediaStorageDir.getPath() + File.  
    separator + "IMG_" + timeStamp + ".jpg");  
FileOutputStream fos = new FileOutputStream(pictureFile);  
fos.write(byteArray);  
fos.close();
```

---

Výpis 23: Vytvorenie a zápis mediálneho súboru

Ďalšou z veľkých výhod použitia fragmentu v procese ukladania súboru je, že vývojár má kontrolu nad tým čo sa deje pred, počas aj po vykonaní metódy bežiacej v samostatnom vlákne. Použitie metódy `onPostExecute()` je ideálne na zobrazenie progress dialógu, aby bol používateľ informovaný o tom, že v aplikácii práve prebieha nejaký proces a on ako používateľ by mal počkať, až bude tento proces dokončený.



Obr. 18: Dialóg progresu procesu

Vo chvíli dokončenia metódy `doInBackground()` je spustená metóda `onPostExecute()` ktorá je využitá pre informovanie používateľa, že daný proces ukladania fotografie bol úspešne dokončený.

---

```

@Override
protected void onPreExecute() {
    super.onPreExecute();
    if(mListener != null)
        mListener.showDialog("Saving...");
}

@Override
protected void onPostExecute(Boolean aBoolean) {
    super.onPostExecute(aBoolean);
    if(mListener != null)
        mListener.onSaveCompleted(aBoolean);
    Toast toast = Toast.makeText(getActivity(), "Picture saved", Toast.
        LENGTH_LONG);
    toast.show();
}

```

---

Výpis 24: Metódy počas a po dokončení vlákna

## 9.8 Prezeranie vytvorených snímok v galérii

Používateľovi je umožnené v aplikácii Camera3D otvorenie ním preferovanej galérie a prezera-  
nie vytvorených fotografií uložených v zložke nazvanej "Camera3D". Bol nastavený listener na  
tlačidlo, ktoré spustí aktivitu galérie a v intente jej predá všetky podstatné informácie ako v  
následujúcom zdrojovom kóde.



---

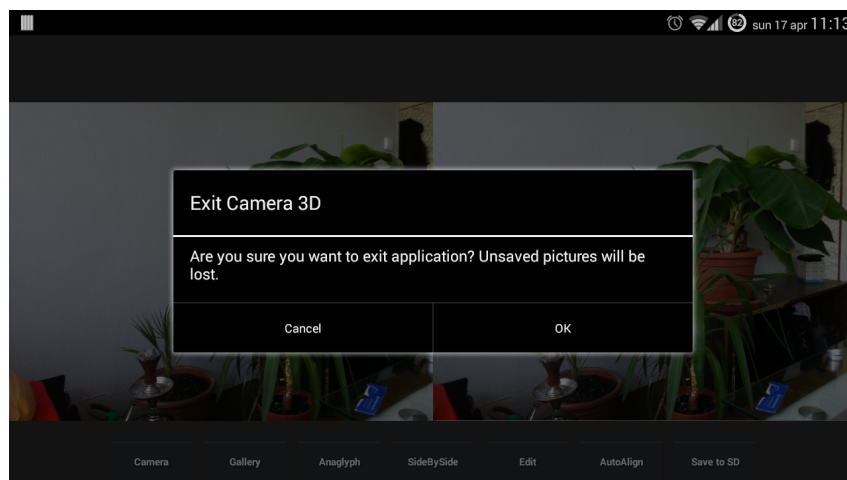
```
Button btnGallery = (Button) layout.findViewById(R.id.button_openGalery);
btnGallery.setOnClickListener(new View.OnClickListener(){
    public void onClick(View view){
        Intent intent = new Intent();
        intent.setAction(android.content.Intent.ACTION_VIEW);
        intent.setType("image/*");
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    }
});
```

---

Výpis 25: Otvorenie galérie

## 9.9 Varovné dialógy

Neoddeliteľnou súčasťou aplikácie sú varovné dialógy vďaka ktorým je používateľ upozornený, že sa chystá vykonať krok kvôli ktorému môže nenávratne stratiť jeho doterajšiu prácu v aplikácii, napr. v prípade, kedy chce vytvoriť novú 3D snímku pričom aktuálnu snímku ešte neuložil, alebo sa chystá bez uloženia ukončiť aplikáciu.



Obr. 19: Varovný dialóg

Zdrojový kód používania dialógov nie je vôbec zložitý.

---

```
AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
builder.setTitle("Exit Camera 3D");
builder.setMessage("Are you sure you want to exit application? Unsaved
    pictures will be lost.")
```

---

Výpis 26: Definícia dialógu

---

```
.setPositiveButton("OK", new DialogInterface.OnClickListener() {
```

---

Výpis 27: Nastavenie čo sa stane po stlačení tlačidla OK

---

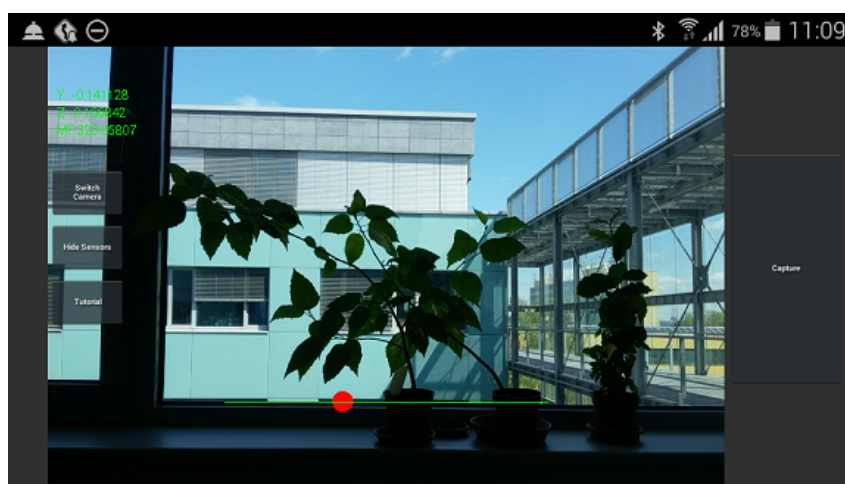
```
.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
```

---

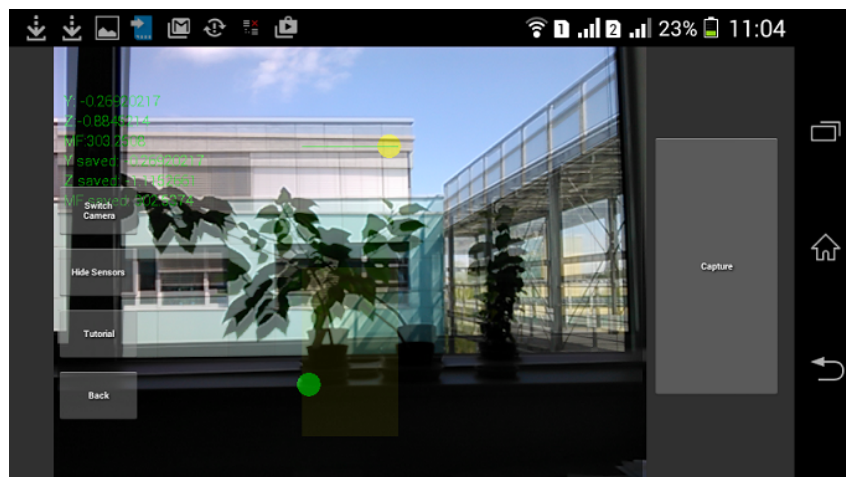
Výpis 28: Nastavenie čo sa stane po stlačení tlačidla Cancel

## 10 Testovanie aplikácie Camera3D

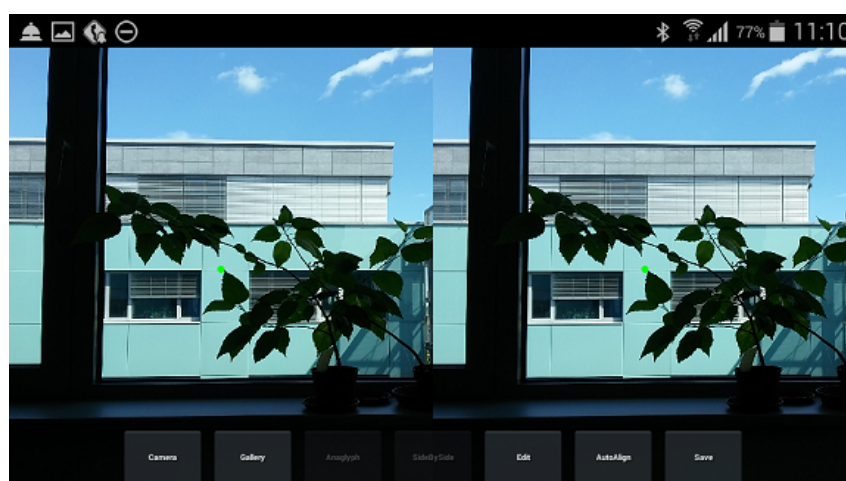
Testovanie prebiehalo na viacerých zariadeniach Samsung Galaxi S3 a S3 mini, S4 mini a Sony Xperia. Testovanie na Android emulátore nebolo možné, keďže hlavnou pointou testovania Aplikácie Camera3D je snímanie fotografií a nakláňanie zariadenia, čo nie je na PC emulátore možné. Aplikácia bezproblémovo funguje na všetkých zariadeniach, avšak bolo by vhodné použitie rozdielnych layoutov pre zariadenia s väčším rozlíšením obrazovky a to Samsung S3 a S4 mini. Napriek tomu, že prvkom užívateľského rozhrania boli pridelené relatívne rozmery v dpi, pri testovaní sa ukázalo, že niektoré prvky sa zobrazujú v neprimerane malých, alebo naopak neprimerane veľkých rozmeroch, avšak stále je to v tolerancií a aplikácia má na všetkých testovaných zariadeniach prívetivé grafické používateľské prostredie. Aplikácia reaguje veľmi svižne a jej stabilita je stopercentná. Nasleduje niekoľko snímok obrazoviek z aplikácie Camera3D vyrobených na rôznych zariadeniach.



Obr. 20: SAMSUNG S4 mini - snímanie prvej fotografie



Obr. 21: SONY Xperia M - snímání druhej fotografie



Obr. 22: SAMSUNG S4 mini - funkcia polo-automatického spojenia



Obr. 23: SAMSUNG S3 - vytvorenie anaglyfu

## 11 Porovnanie aplikácie Camera3D s existujúcimi aplikáciami

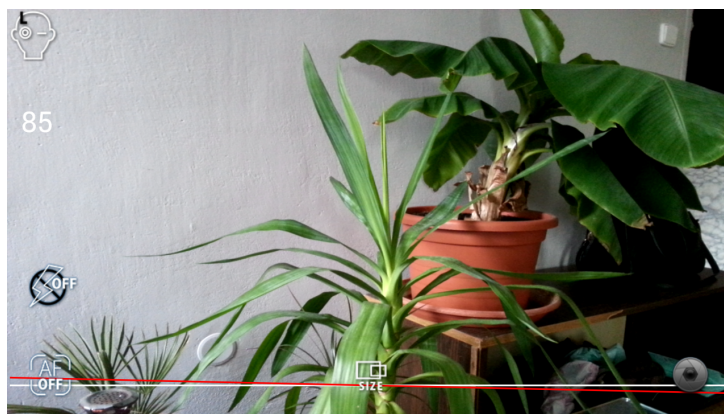
Nie vždy sa dajú presne odhadnúť potreby a túžby používateľov Android aplikácií, a preto sa nedá bez širšieho prieskumu na používateľoch odhadnúť, ktorá z aplikácií na snímanie 3D fotografií by bola najviac obľúbená. Avšak, je možné porovnávať funkcie novo vyvinutej aplikácie s jednotlivými funkciami existujúcich aplikácií, a podľa tohto porovnania odhadnúť potenciálny úspech novej aplikácie. Z tabuľky 2 jasne vyplýva, že v aplikácii Camera3D sú implementované všetky známe funkcie. Nasleduje porovnanie niektorých vybraných funkcií aplikácie Camera3D v porovnaní s prieskumom z kapitoly 5.2.

Aplikácia > Funkcia	Camera3D
Aspoň dve rôzne vizualizačné techniky	Áno
Anaglyf	Áno
Senzorické navádzanie	Áno
Korekcia naklonenia	Áno
Korekcia rotácie	Áno
Manuálne spojenie snímok	Áno
Poloautomatické spojenie snímok	Áno
Fotografovanie prednou kamerou	Áno
Transparentné zobrazenie prvej fotografie pri snímaní druhej fotografie	Áno
Reálne zobrazenie pomeru strán prvej fotografie pri snímaní druhej fotografie	Áno
Zobrazuje reklamy	Nie
Najvyššie rozlíšenie snímok	2048/1536 3.1MP

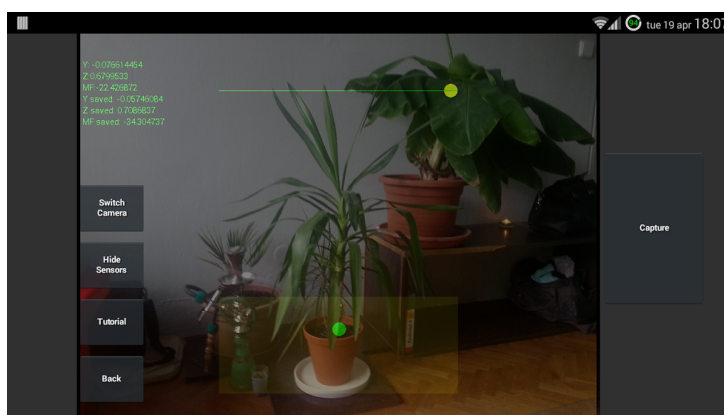
Tabuľka 2: Výpis funkcií aplikácie Camera3D

### 11.1 Indikácia vhodného okamžiku pri snímaní druhej fotografie

Aplikácia 3D Steroid ako jediná z prieskumu, pomáha používateľovi pri snímaní fotografie, a to tak, že graficky zobrazuje pozíciu zariadenia na horizontálnej ose. Avšak novo vyvinutá aplikácia Camera3D zaznamenáva naklonenia a rotácie vo všetkých smeroch 25 a pri snímaní druhej fotografie graficky indikuje okamih vhodný na snímanie, čo je veľmi komplexný proces a už vďaka tejto funkcií aplikácia Camera3D prekonáva ostatné aplikácie.



Obr. 24: 3D Steroid grafické zobrazenie pozície na ose X



Obr. 25: Camera3D indikácia vhodného okamžiku snímania pomocou údajov zo senzorov na osách X,Y,Z

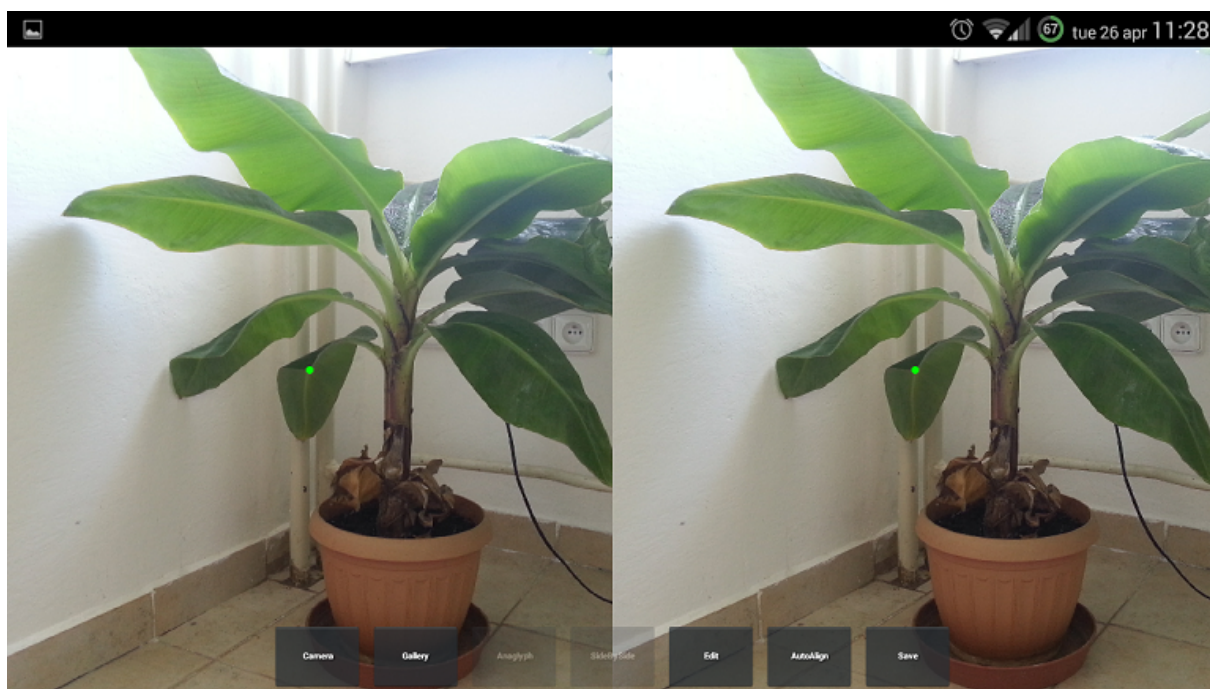
## 11.2 Najvyššie rozlíšenie snímok

Ak je fotoaparát zariadenia dostatočne kvalitný, aplikácia Camera3D ukladá snímky až v kvalite 3.1MP čo je výrazne zvýšenie kvality v porovnaní s ostatnými existujúcimi aplikáciami. Výpis najvyšších rozlíšení existujúcich aplikácií bol uvedený v tabuľke 1 a žiadna z aplikácií neprekročila hranicu 0.8MP.

## 11.3 Polo-automatické spojenie snímok

Medzi aplikáciami z prieskumu nie je žiadna podobná aplikácia, ktorá by implementovala funkciu polo-automatického spojenia, ktorú predstavuje aplikácia Camera3D. Vďaka používateľovej identifikácii spoločných bodov je do určitej miery táto metóda rýchla, pretože vyberanie jedného konkrétneho bodu je pre používateľa elementárna operácia, a zároveň nepridáva žiadny výpočtový čas do procesu spojenia fotografií. Táto metóda je veľmi presná, a funguje tak ako je zobrazené na obrázku 26.





Obr. 26: Funkcia polo-automatického spojenia snímok



Obr. 27: Vľavo:bez použitia metódy polo-automatického spojenia, vpravo: s použitím metódy

## 12 Záver

Cieľom diplomovej práce bolo navrhnutie a vývoj mobilnej aplikácie a knižnice pre snímanie 3D fotografií, ktorý sa úspešne podarilo splniť. Dôvodom vývoja aplikácie Camera3D bola snaha priniesť aplikáciu s novou a vylepšenou funkcionalitou, ktorá by prekonala v súčasnosti dostupné aplikácie pre mobilné zariadenia.

Vyvinutej aplikácií bol udelený názov Camera3D a s využitím implementovanej vlastnej knižnice priniesla úplne novú a do súčasnosti nikde inde nepoužitú funkciu polo-automatického spojenia stereoskopického páru na základe zhodného bodu snímok, určeného používateľom. Táto funkcia bola pri návrhu aplikácie vybraná ako ideálny nástroj spojenia stereoskopického páru z dôvodu potenciálne veľkej presnosti spojenia a zároveň zachovania jednoduchosti a rýchlosti aplikácie. Aplikácia Camera3D súčasne priniesla funkciu indikácie vhodného okamžiku snímania druhej fotografie stereoskopického páru pomocou rotačných senzorov mobilného zariadenia, na základe rotácií zariadenia vo všetkých smeroch otáčania.

Aplikácia Camera3D ponúka zvýšenú kvalitu do tejto doby poznaných funkcií, ako ukladanie fotografií až v trojnásobne vyššom rozlíšení, alebo zobrazenie náhľadu v reálnom pomere strán fotografie pri snímaní. Na niektorých funkciách existujúcich aplikácií už nebolo čo vylepšovať a tak boli implementované bez výrazných vylepšení, avšak do súčasnosti neexistovala aplikácia, ktorý by ponúkala kombináciu všetkých funkcií uvedených v tejto práci zároveň, ako snímanie predným fotoaparátom, snímanie na výšku s korekciou rotácie, vytváranie viacerých vizualizačných formátov alebo manuálne spojenie snímok. Na záver boli aplikácia Camera3D a jej funkcie podrobne porovnané s existujúcimi aplikáciami pričom vo po všetkých stránkach Camera3D bola aspoň tak dobrá ako najlepšia aplikácia vybraná podľa najlepšej použitej funkcie.

Aplikácia Camera3D úspešne prešla testovaním stability, rýchlosti a celkovej kvality spracovania 3D snímok. v prílohe diplomovej práce sú dostupné ukážky 3D fotografií vyfotografovaných a upravených aplikáciou Camera3D B.



## Literatúra

- [1] Stidwill, D., and Fletcher, R. (2010). Normal binocular vision: Theory, investigation and practical aspects. John Wiley and Sons
- [2] Robitza-3d-vision.pdf [online]. [cit. 6.2.2016]. Dostupné z: <http://slhck.info/documents/robitza-3d-vision.pdf>
- [3] HROMÁDKOVÁ, Lada, 2011. Šilhání. Brno: Národní centrum ošetrovatelství a nelékařských zdravotnických oborů. ISBN 9788070135303 8070135301.
- [4] HLOUŠKOVÁ, Iva. Vliv 3D technologií na vidění [online]. [cit. 6.2.2016]. Dostupné z: <https://theses.cz/id/c09ygv/>
- [5] Klúče vnímania hĺbký a priestoru [online]. [cit. 6.2.2016]. Dostupné z: <http://www.percepcia.szm.com/hlbka.htm/>
- [6] Stereofotografie [online]. [cit. 6.2.2016]. Dostupné z: <http://klub.stereofotograf.eu/index.php>
- [7] Matěj Boháč, Zpusob zobrazení prostoru [online]. [cit. 6.2.2016]. Dostupné z: [http://www.kiv.zcu.cz/lobaz/uf3d/01\\_uvod/text01\\_uvod.html](http://www.kiv.zcu.cz/lobaz/uf3d/01_uvod/text01_uvod.html)
- [8] Míra prostorovosti [online]. [cit. 6.2.2016]. Dostupné z: [http://klub.stereofotograf.eu/deviace\\_1.php/](http://klub.stereofotograf.eu/deviace_1.php/)
- [9] 3D fotografie a alternativní techniky ve fotografii [online]. [cit. 6.2.2016]. Dostupné z: [http://www.kiv.zcu.cz/lobaz/uf3d/01\\_uvod/text01\\_uvod.html/](http://www.kiv.zcu.cz/lobaz/uf3d/01_uvod/text01_uvod.html/)
- [10] Matěj Boháč, Způsob zobrazení prostoru [online]. [cit. 6.2.2016]. Dostupné z: [http://www.kiv.zcu.cz/lobaz/uf3d/01\\_uvod/text01\\_uvod.html/](http://www.kiv.zcu.cz/lobaz/uf3d/01_uvod/text01_uvod.html/)
- [11] Anaglyph glasses test [online]. [cit. 10.2.2016]. Dostupné z: <http://www.david-romeuf.fr/3D/Anaglyphes/BonCoupleEL/GoodCoupleMonitorGlassesAnaglyph.html>
- [12] Google Calling: Inside Android, the gPhone SDK [online]. [cit. 2016-04-10]. Dostupné z: <http://www.onlamp.com/pub/a/onlamp/2007/11/12/google-calling-inside-the-gphone-sdk.html>
- [13] Matěj Boháč, Způsoby snímání stereofotografií [online]. [cit. 10.2.2016]. Dostupné z: <http://klub.stereofotograf.eu/foceni.php>
- [14] Simulated 3D—Wiggle 3D [online]. [cit. 2016-04-10]. Dostupné z: <http://www.shortcourses.com/stereo/stereo1-17.html>
- [15] Podrobný 3D Průvodce: Jak dělat 3D fotografie a 3D videa [online]. 2013 [cit. 2016-04-10]. Dostupné z: [http://www.3djournal.com/ebooks/pruvodce\\_3D\\_foto\\_video.pdf](http://www.3djournal.com/ebooks/pruvodce_3D_foto_video.pdf)

- [16] Stereoscope [online]. [cit. 2016-04-10]. Dostupné z: [http://timbreproductions.com/pages/commentary\\_\\_s](http://timbreproductions.com/pages/commentary__s)
- [17] 3D AUTO-STEREOSKOPICKÉ MONITORY [online]. [cit. 2016-04-10]. Dostupné z: <http://cs.gali-3d.com/autostereoskopie-3d/>
- [18] How IMAX Works [online]. [cit. 2016-04-10]. Dostupné z: <http://entertainment.howstuffworks.com/imax1.htm>
- [19] Google mění Android: přepněte si na ART [online]. [cit. 2016-04-10]. Dostupné z: <http://www.mobilmania.cz/bleskovky/googlemeniandroidprepnetesinaart/sc4a-1325316/default.aspx>
- [20] Java SE Documentation [online]. [cit. 2016-04-10]. Dostupné z: <http://docs.oracle.com/javase/7/docs/technotes/guides/jni/spec/intro.html>
- [21] Taking Photos Simply [online]. [cit. 2016-04-10]. Dostupné z: <http://developer.android.com/training/camera/photobasics.html>
- [22] A PERFORMANCE COMPARISON BETWEEN JAVA AND C ON THE NEXUS 5 [online]. 2014 [cit. 2016-04-10]. Dostupné z: <http://www.learnopengles.com/a-performance-comparison-between-java-and-c-on-the-nexus-5/>
- [23] Směrování fotoaparátů při fotografování [online]. [cit. 2016-04-10]. Dostupné z: [http://klub.stereofotograf.eu/toe\\_in.php](http://klub.stereofotograf.eu/toe_in.php)
- [24] BAY, Herbert, Andreas ESS, Tinne TUYTELAARS a Luc VAN GOOL. Speeded-Up Robust Features (SURF) [ftp://ftp.vision.ee.ethz.ch/publications/articles/]. Zurich Switzerland [cit. 2016-04-26].
- [25] G. LOWE, David. Distinctive Image Features from Scale-Invariant Keypoints. Vancouver, B.C., Canada, 2004. Computer Science Department University of British Columbia.
- [26] Anaglyph Methods Comparison [online]. [cit. 2016-04-10]. Dostupné z: [http://www.3dtv.at/knowhow/anaglyphcomparison\\_en.aspx](http://www.3dtv.at/knowhow/anaglyphcomparison_en.aspx)
- [27] Have Google Cardboard? Here are the best apps that work with it [http://www.gizmotimes.com/apps/google-play/google-]. 2016 [cit. 2016-04-27].

## A Zdrojové kódy

Zdrojový kód orezávania bitmáp na základe používateľovho vzájomného posunu zobrazených obrazov.

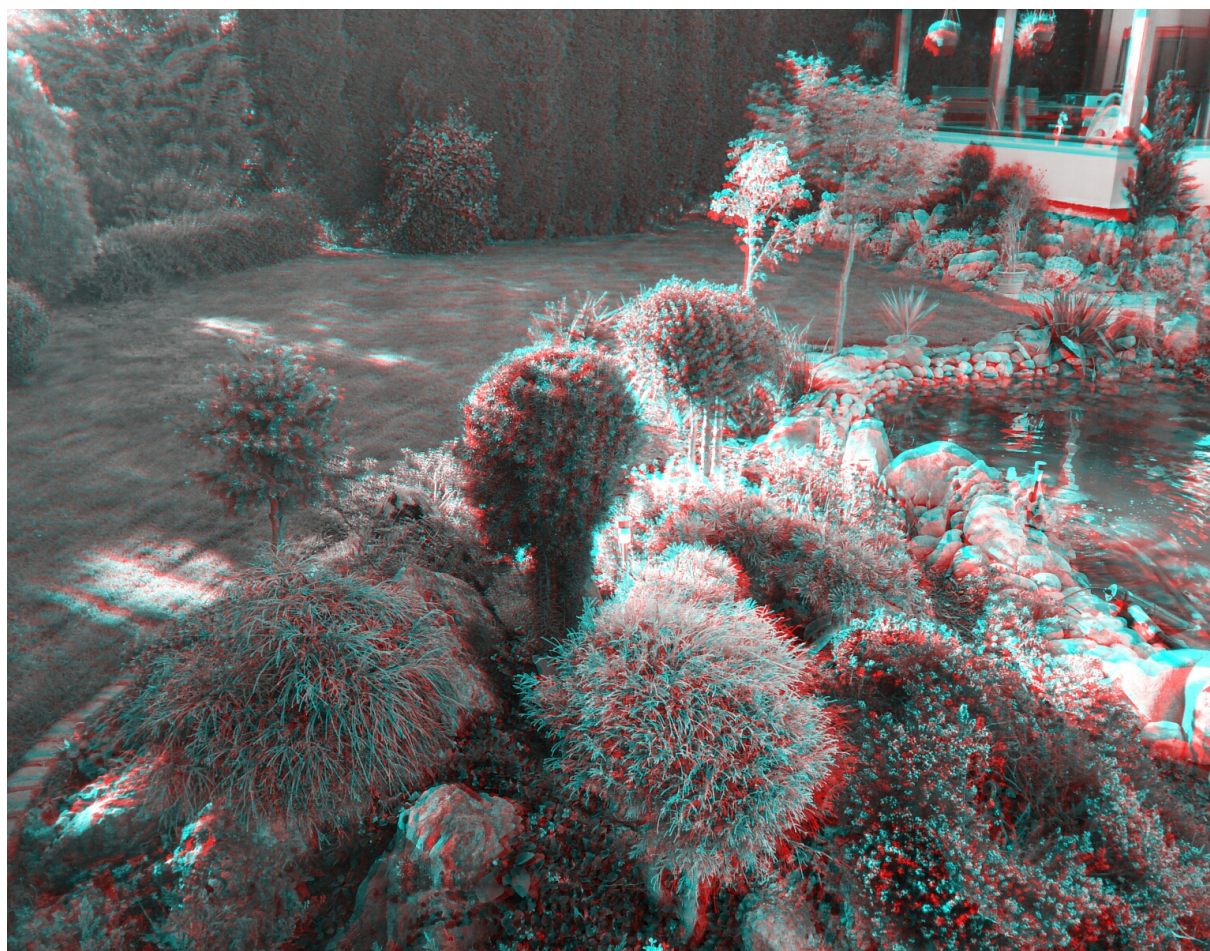
---

```
if(diffLeft >= 0 && diffTop >= 0) {
    bitmapStatic = Bitmap.createBitmap(bitmaps[0], diffLeft, diffTop, width
        - diffLeft, height - diffTop);
    bitmapMoved = Bitmap.createBitmap(bitmaps[1], 0, 0, width - diffLeft,
        height - diffTop);
}else if(diffLeft >= 0 && diffTop <= 0){
    bitmapStatic = Bitmap.createBitmap(bitmaps[0], diffLeft, 0, width -
        diffLeft, height + diffTop);
    bitmapMoved = Bitmap.createBitmap(bitmaps[1], 0, -diffTop, width -
        diffLeft, height + diffTop);
}else if(diffLeft <= 0 && diffTop >= 0) {
    bitmapStatic = Bitmap.createBitmap(bitmaps[0], 0, diffTop, width +
        diffLeft, height - diffTop);
    bitmapMoved = Bitmap.createBitmap(bitmaps[1], -diffLeft, diffTop, width
        + diffLeft, height - diffTop);
}else {
    bitmapStatic = Bitmap.createBitmap(bitmaps[0], 0, 0, width + diffLeft,
        height + diffTop);
    bitmapMoved = Bitmap.createBitmap(bitmaps[1], -diffLeft, -diffTop,
        width + diffLeft, height + diffTop);
}
```

---

Výpis 29: Orežanie bitmáp v prospech spojenia snímok

## B Ukázkové obrázky vyrobené a upravené aplikáciou Camera3D



Obr. 28: Ukážka anaglyfu





Obr. 29: Ukážka Side-By-Side 1.



Obr. 30: Ukážka Side-By-Side 2. - vytvorená na výšku